FLORIDA INTERNATIONAL UNIVERSITY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

EEL-4920 – SENIOR DESIGN I

Summer 2022

Stock Market Trader Bot

Team 15

Daniel Leal

Joseph Dabreu

Gregory Hill

Eddie Lezama

SENIOR DESIGN II PROPOSAL

Submitted to:

Senior Design I Instructor: Professor Wilmer Arellano

Mentor: Associate Teaching Professor, Ph.D., Wilmer Arellano

July 2022

TABLE OF CONTENTS

Title Page……..I
Table of Contents……II
List of Tables……..IV
List of Figures…….....V
Acknowledgement……VII
Abstract……...VIII

I. Executive Summary……..1
II. Problem Statement…….....4
A. Objectives……...4
B. Constraints…….....5
III. Assumptions and Limitations……6
A. Assumptions……...6
B. Limitations…….....6
IV. Needs Feasibility Analysis……7
A. Needs Analysis……...7
B. Needs Specification…...……10
C. Feasibility Analysis……...12
D. Marketability…….....19
V. Risk Analysis……26
VI. Operating Environment……...30
VII. Intended User(s) and Intended Use(s) ……...31
A. Intended User(s) ……...31
B. Intended Use(s) ……....31
VIII. Background……...32
IX. Intellectual Property Considerations……41
X. Globalization…….....46
XI. Standards Consideration……..50
XII. Health and Safety Considerations……..55
XIII. Environmental Considerations……..57

XIV. Sustainability Considerations……...60
XV. Manufacturability Considerations……..62
XVI. Ethical Considerations and Social Impact…..……..64
A. Ethical Considerations……..64
B. Social Impact……....66
XVII. Concept Development…….....67
A. Alternative Options……...67
1) Advantages……67
2) Disadvantages……68
XVIII. End Product Description and Other Deliverables……..74
A. End Product Description……...74
B. Functions……...76
C. Specifications……80
D. Other Deliverables……80
XIX. Plan of Action……...81
A. Statement of Work (SOW) ……...81
B. Work Breakdown Structure……...82
C. Project Milestones…….....85
D. Gantt Charts……..85
XX. Multidisciplinary Aspects……..87
XXI. Personnel……..89
XXII. Budget…….93
XXIII. Results Evaluation…….95
XXIV. Life-Long Learning……...98
XXV. Conclusion……..99
XXVI. References……100
XXVII. Appendices……105
A. Team Contract……105
B. Intellectual Property Contract……106
C. Provisional Patent Form……..107
XXVII. Signatures Page……...109

LIST OF TABLES

Table I.	Objectives	7
Table II.	Constraints	8
Table III.	Assumptions	8
Table IV.	Limitations	8
Table V.	Project Objectives	9
Table VI.	Final Project Objectives	10
Table VII.	Specifications	11
Table VIII.	Technical Feasibility Assessment	13
Table IX.	Resource Feasibility Assessment	14
Table X.	Schedule Feasibility Assessment	14
Table XI.	Economic Feasibility Assessment	15
Table XII.	Cultural Feasibility Assessment	16
Table XIII.	Legal Feasibility Assessment	17
Table XIV.	Market Feasibility Assessment	17
Table XV.	Feasibility Weighting	18
Table XVI.	Weighted Feasibility Score	19
Table XVII.	Zeke Specifications	21
Table XVIII.	Finnhub Feature, Purpose, and Functionality	24
Table XIX.	Risk Matrix	28
Table XX.	Risk Mitigation Actions	28
Table XXI.	Solutions for Ethical Dilemmas	65
Table XXII.	Weight of Ethical Theories	66
Table XXIII.	Mean Square Error Form	69
Table XXIV.	Weight Calculation Table	73
Table XXV.	Concept Selection Table	73
Table XXVI.	Functionality of Machine Learning	77
Table XXVII.	Functionality of Reinforcement Learning Algorithms	78
Table XXVIII.	Functionality of Bollinger Bands	79
Table XXIX.	Specifications	80

LIST OF FIGURES

Figure 1.	Fishbone Diagram	11
Figure 2.	Zeke Mobile App	20
Figure 3.	Zeke Trigger Page UI	22
Figure 4.	Zeke Functionality Block Diagram	22
Figure 5.	Finnhub's Home Page	23
Figure 6.	REST API Architecture	24
Figure 7.	Fishbone Diagram of Risk Factors	26
Figure 8.	The Computing Inside a Hedge Fund	34
Figure 9.	The Interconnection of a Hedge Fund	34
Figure 10.	The Computing Inside N-Day Forecasts	35
Figure 11.	The Bollinger Bands	37
Figure 12.	Grinold's Fundamental Law	37
Figure 13.	The Capital Asset Pricing Model	38
Figure 14.	The Arbitrage Pricing Theory	38
Figure 15.	Executing Orders Programmatically	40
Figure 16.	Diagram of Patented Function	41
Figure 17.	Diagram of an Example of Patent A	42
Figure 18.	Summary of Process of Trading Financial Instruments	43
Figure 19.	Basic Overview of Electronic Trading Environment System	45
Figure 20.	Software System and Element Relationship	51
Figure 21.	Example of Software System-of-Interest Structure	52
Figure 22.	Relationship Between Maintenance Process and Other…	53
Figure 23.	Verification and Validation	54
Figure 24.	Project Concept Fan	67
Figure 25.	Bar Graph View of MSE Results Across Various Stocks	70
Figure 26.	The Architecture of the Proposed Deep Reinforcement Learning…	71
Figure 27.	End Product Description Overview	74
Figure 28.	General Overview of Automatic Transaction Handling	75
Figure 29.	General Overview of Client/User Intervention Handling	76
Figure 30.	Functionality of Data Science/Machine Learning in the Bot	77

Figure 31.	General Overview of Reinforcement Learning Functionality	78
Figure 32.	General Overview of Bollinger Bands Functionality	79
Figure 33.	Work Breakdown Structure	83
Figure 34.	Task View Grantt Chart	85
Figure 35.	Task-Time Distribution Grantt Chart	86
Figure 36.	PERT Chart	86
Figure 37.	Task View Budget	94
Figure 38.	Resource Cost	94

ACKNOWLEDGEMENT

The participating engineers wish to extendedly express their heartfelt gratitude to those who, in one way or another, provided guidance and assistance throughout the completion of the Stock Market Trader Bot.

The team members unalloyed appreciation is outstretched to our humble and amiable mentor, Professor Dr. Wilmer Arellano for his invaluable contributions, outstanding guidance, and instruction throughout the course of Senior Design.

ABSTRACT

The volatility of the Stock Market can be pushed at a pace that is difficult for the average trader to follow. To guide and assist traders in making the most profitable decisions, the Stock Market Trader Bot, in theory, provides novice traders the opportunity for financial profits at speeds that is near impossible for a human trader. This Bot follows principles of algorithmic trading using programming to create buy/sell orders and automatically submits them to a market center, or exchange. This Bot's end product is envisioned to comply to all artificial intelligence and bot standards through active design scheduling outlined by a plan of action and equal distribution of work based on the multidisciplinary strengths of its engineers. Overall, the Trader Bot will combine economic theories, complex reinforcement learning algorithms, and a coherent, experienced team of computer and electrical engineers to create a Bot with a high trading success measurement.

I. EXECUTIVE SUMMARY

Stock Market Trader Bot	
Team Name: Pythonic Traders	
Team Number: 15	Mentor: Dr. Wilmer Arellano
Team Leader: Daniel Leal	Team Member: Joseph Dabreu
Team Member: Gregory Hill	Team Member: Eddie Lezama

A. Summarized Problem Statement

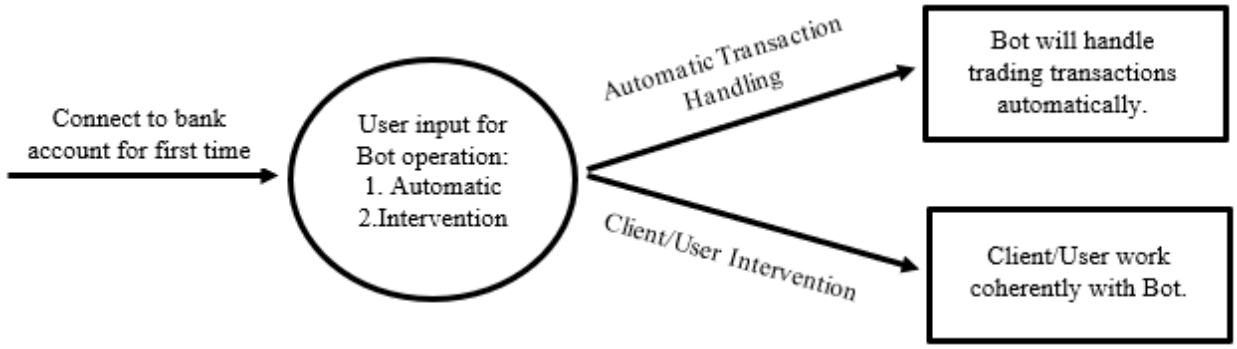
As a direct consequence of the volatile nature of the Stock Market, novice traders are becoming increasingly discouraged to invest and trade. According to data accumulated from the 2002 to 2017, over 75% of all day traders quit within two years [1]. Etoro, a Stock Market Platform, found that over the course of one-year, modern day traders concluded with a median loss of over 30% [2]. Following the daily drastic changes and behaviors of multiple stocks in a personal portfolio can encourage monetary loses and declined motivation to invest in varying exchanges.

B. Objectives and Constraints

In summary, the end product is envisioned to comply to specific, detailed objectives and constraints. The specific objectives can be classified into main categorical objectives that serve to summarize the overall goals. Such main objectives include safety, progressive training, marketability, and user-friendliness. For the specific constraints, the Bot is created to be compatible with standard computers, be operable cross-platform, and possess the ability to operate as a simulator for potential transactions.

C. Project Description

The Stock Market Trader Bot will have a clear user interface that allows for easy handling of the Bot's functional trading operations. Prior to any trading, the system will require a direct and established connection to a properly funded, compatible bank to allow for legal and effective connection, ensuring the ability to manipulate buy/sell orders. From this friendly interface, the client will be able to select whether it wants its current portfolio to be handled completely by the Bot using the Automatic Transaction Handling option or wants to work alongside the Bot using the Client/User Intervention option as shown in the following image from the End Product section:



*D. **Sections***

1) Background

Prior to deciding how the end product would be accomplished, extensive research was performed in the fields related to the Stock Market Trader Bot to provide clear context as to not only the related projects currently in market, but what theories, approaches, and conceptual analysis was implemented and utilized by these analogous products. This section depicts a story as to how the evolution of the different aspects of the Stock Market Trader Bot came to be

a) Hedge Fund and Machine Learning Computing

The computing behind Hedge Funds allows for the appropriate framework and idea introduction as to how the Bot can apply machine learning to simulate the actions of a trader. More specifically, machine learning algorithms such as k-nearest neighbor can be paired with historical pricing data of individual stocks to formulate a predictive model that analyzes trends to understand the current position of each stock. By training this model, the algorithm can accurately forecast the future and apply trading algorithms to optimize a portfolio and pursue an automated profitable experience to users.

b) Bollinger Bands and Grinold's Fundamental Law

Bollinger Bands utilizes a rolling mean, or a standard moving average, of varying data points of a stock's closing price. Above and below this standard moving average, an upper and lower band is plotted from the standard deviation of this rolling mean. From these Bollinger Bands, a visual and statistical analysis can be performed to understand if a stock is of a good price to buy or sell depending on its original closing price line intersection with the upper or lower Bollinger Band. Moreover, Grinold's Fundamental Law is widely used to validate adding securities to an optimization universe, adding factors to forecast return, trading frequently, and reducing constraints [10]. These both allow for an easier Bot decision-making process for buy/sell orders.

c) Interactive Brokers (IB)

Interactive Brokers is an American multinational brokerage firm. IB operates the largest electronic trading platform in the country of the United States [16]. IB is relevant to the field of algorithmic trading because it also serves as a programming interface for executing orders. However, IB stands out from its peers because of its mature, flexible, and high-speed API and its significantly lower commissions in comparison to other trading firms.

2) End Product Description

The End Product Description explains how the Stock Market Trader Bot will operate along with the functions it will perform and can be categorized by its two functional operations: 1) Automatic Transaction Handline and 2) Client/User Intervention

a) Automatic Transaction Handling

For this form of operation, the Bot will proceed to perform buy/sell orders of a predetermined portfolio of varying stocks based off algorithmically driven decision-making models. This specific option of operation for the Bot does not require any user intervention other than selecting a pool of stocks in which the user wishes to allow the Bot to manipulate through varying orders and exchanges. From here on forth, the Bot will use the library of machine learning algorithms it has been trained with to predict a stock's price based on historical data analysis ranging from approximately 25 years prior. Furthermore, the Bot will also use economic theories to formulate a trend and moving average to understand the behavior of a select stock.

b) Client/User Intervention

If the client was to select the alternative option, "Client/User Intervention," the Bot continues to apply all its machine learning models, historical data analysis, Q Learning/Dyna algorithms, and economic theories. However, rather than automatically handling buy/sell orders and the decision-making process when controlling stock orders in a portfolio, this option will leave the choice to the user. In this functional option, the primary position and tasks of the system is to merely provide suggestions and notifications to the user of when it recommends acting on an investment.

3) Budget

With the knowledge gained from research and priorly obtained multidisciplinary skills, the team will be able to produce a software bot that can predict outcomes of future stock prices by algorithmically analyzing past data. Due to the project being software labor intensive and requiring no hardware or material cost to be produced, there will be no outstanding cost to the project itself. However, an evaluation must be made to decide how much the project will cost as a result of all the hours of labor the team members must perform to compose of the software. More specifically, files presented in the plan of action and budget sections, the estimated total cost of the project purely because of its labor and time intensity, will be $62469.96. Lastly, expenses to take into consideration are books and courses that the team members mutually agree to purchase for beneficial production purposes.

*E. **Conclusion***

The engineers of this team have always enchanted spiked interest in a product that involved the opportunity for monetary profitability whilst implementing principles of computer and electrical engineering undergraduate studies. Because of this common ground, the Stock Market Trader Bot began to be constructed. In the design phase, the most important objectives include ensuring the product will have a success measurement high enough for the product to be generalized as profitable, will have the ability to opt-in and provide two functional options for trading (Automatic Transaction Handling or Client/User Intervention), and will be easy to utilize and maneuver by the user with minimal trading experience. These objectives resulted from the client interview performed and brainstorming amongst team members. Furthermore, the composition of this design phase involved a variety of activities such as team brainstorming of potential projects, consulting with the team mentor regarding the best overall project, understanding all aspects of the project, in-depth research of the aimed end product, client interview, and end-product deliverables. Overall, the main goal for the product is to provide a high chance of gaining pecuniary profit with minimal prior experiencing by the end of the implementation phase (Senior Design II) of the Stock Trader Bot. To do so, constraints such as possessing the capability to operate as a simulator for potential transactions, comparability with standard computers, and ensuring cross-platform abilities must also be met. Lastly, as outlined in the results evaluation and deliverables section, the Stock Trader Bot will be designed and implemented in a manner that complies with all AI and IEEE standards for proper publication. By following through with all claims and goals outlined in the prior sections, the project is sure to positively contribute to society in a global manner by providing its users monetary earnings with little to no applied effort. In conclusion, the entirety of the process pertaining to the designing of the Stock Market Trader Bot has contributed to significant lifelong learnings for all team members in fields of high demand such as machine learning, artificial intelligence, software development in a team environment, economics and finance, and data analytics through statistical and programming applications.

II. PROBLEM STATEMENT

Due to the extreme nature of the Stock Market, novice traders are becoming increasingly discouraged to actively invest and trade. As of October 2017, according to transaction data accumulated from the entirety of the 15 years prior, it was evident that more than 75% of all day traders quit within two years, and "poor performers were most likely to quit" [1]. Furthermore, according to the stock platform Etoro, it was found that over the course of one year, day traders concluded with a median loss of over 30% [2]. Following the daily drastic changes and behaviors of multiple stocks in an individual's portfolio can inspirit and embolden consistent monetary loses.

The proposed product, Stock Market Trader Bot, aims to produce consistent pecuniary gains at velocities that is increasingly strenuous for a human trader. Overall, the primary purpose for this Bot is to appropriately govern and control risk while expanding and enlarging profitability in any market environment for its current user. Additionally, the Stock Market Trader Bot prioritizes a user-friendly atmosphere, a modular product, easy intervention of the user trader pertaining to buy/sell orders and a high success measurement to be considered profitable.

In this problem statement section, crucial information will be highlighted pertaining to how engineering principles and procedures will be applied to the product. This section is important because once a problem has been identified, objectives and constraints must be clearly outlined to organize a sophisticated and intricate engineering design and implementation process. This product will effectively serve as a solution to the issue and problem statement at hand.

*A. **Objectives***

1. Safety

1.1 The Stock Market Trader Bot is safe to use for all users with minimal financial knowledge, trading experience, and technical abilities.

1.2 The Stock Market Trader Bot will be safe and reliable to use cross-platform.

2. Progressive Training

2.1 The Stock Market Trader Bot aims to profusely analyze the behavior of the market and eventually anticipate changes, allowing for this product to, in the future, become accessible to all individuals interested.

2.2 The Stock Market Trader Bot is fully modular, meaning the user interface with respect to software updates are clear and client friendly.

3. Marketability

3.1 The Stock Market Trader Bot will have a success measurement high enough for the product to be generalized as profitable.

4. User-Friendly

4.1 The Stock Market Trader Bot will have the ability to opt-in and provide two functional options for trading: 1) Automatic Transaction Handling and 2) Client/User Intervention

4.2 The Stock Market Trader Bot provides an easily manipulated and controllable user interface to facilitate functional intervention.

4.3 The Stock Market Trader Bot will provide novice traders the opportunity of a profitable experience in the stock market and trade at a high performing level with minimal user effort and activity.

4.4 The Stock Market Trader Bot will be easy to utilize and maneuver by the user with minimal prior trading and stock exchange background.

*B. **Constraints***

1. The Stock Market Trader Bot will have no material cost and must work in standard computers.

2. The Stock Market Trader Bot must be cross-platform (Linux, Macintosh, & Windows).

3. The Stock Market Trader Bot must possess the capability to operate as a simulator for potential transactions.

III. ASSUMPTIONS AND LIMITATIONS

To ensure a smooth and cultivated product implementation and creation process, explicit and precise assumptions and limitations must be outlined. These product-specific details will play a large factor into the approach taken by the stake holding engineers in the design and execution phases of this project. Both the assumptions and limitations should proactively conduct and direct the engineers into the proper framework, or specifications, that is wished to be produced by the final product. More specifically, in this section, the most critical instructions, regulations, and requirements will be defined to include the most critical features that the Stock Market Trader Bot is desired and fancied to possess.

Assumptions, pertaining to this contemporary product, are circumstances that are assumed to be implemented, or featured, into the overall final product. Limitations are, in comparison to assumptions, more rigid constraints either inflicted by the client/user, or by the unpliable technological and physical limitations relevant to the particular technology that is being utilized (e.g., amount of memory in computer occupied by the downloadable algorithm/product). Furthermore, the limitations defined accurately represents conditions that, should they not be met, will encourage a failed project objectives and potentially eliminating partial usefulness of the bot.

A. Assumptions

1. The designers assume that the Stock Market Trader Bot to only have appropriate and sufficient capabilities to analyze trade stocks that is visible to the public and the average trader.
2. The Stock Market Trader Bot assumes to be primarily confined to the United States Stock Market and potentially be implementable to those actively participating in a foreign exchange market (e.g., Taiwan Exchange Market).
3. The Stock Market Trader Bot assumes to have two primary options with respect to client usage: Manual operation (User approves execution for buy/sell orders of trades) and Automatic operation (Stock Market Trader Bot would place the buy/sell orders of trades without user intervention).

B. Limitations

1. Functionality of the Stock Market Trader Bot in the “Manual” option of the user interface requires the user to have preexisting knowledge of stock exchanges to appropriately tell the bot where to focus its priority for buy/sell orders.
2. The Stock Market Trader Bot is not responsible for monetary loss and the user assumes all risk and responsibility with respect to personal pecuniary change resulting from the bot’s investment decisions.

In conclusion, the assumptions outlined ensure that clear expectations are provided as to how the product is planned to operate along with a brief description as to the specific environment the Bot will operate in. Assumptions are crucial to the envisioned end product and are presumed to be firmly implemented. On the other hand, the limitations subsection shows that there might be potential bumps when the implementation process is set to begin. However, to conclude, the participating engineers will guarantee that assumptions are met and that limitations will be minimized to the maximum ability expected.

IV. NEEDS FEASIBLITY ANALYSIS

In this section, we will provide a Needs and Feasibility analysis. Specific to the project, the output of the Needs Analysis results in the project's objectives, client's constraints, design assumptions, and limitations. The design team next defines the project specifications necessary to build the project. The specifications technically define the project's success criteria for meeting the project's objectives. This analysis consists of the Needs Analysis, Specifications definitions, and a Marketability review of the project. The Needs Analysis is the output of the client interview completed with the Faculty Advisor. Following the Client Interview, the team conducted a brainstorming session to review the client's input and identify any additional objectives, assumptions, or limitations not recognized by the client.

Finally, this section concludes with a Marketability review of the project. The purpose of which is to review comparable products available on the market. We observe commonalities between the projects and similar products. These comparisons form the basis of potential pricing decisions for the project. Additionally, we ensure we have designed a competitive product following all ethical guidelines.

A. Needs Analysis

The Needs analysis begins with a client interview and concludes with drafting a problem statement and objectives. The design team met with the client and faculty advisor to review his goals for the project. In the initial discussion, we understood the desired system would be an algorithmic trading system designed for use with the Interactive Broker. The client interview provided the design team insight into the desired end product from the client's perspective. We were also able to clarify the intended users of the system and the ideal operation. The design team creates a problem statement based on the initial client interview. Table I lists the objectives captured from the client meeting.

1) Client Interview

TABLE I. OBJECTIVES

Client Interview	
Source	***Objective***
Client	The system should be an algorithmic trader
Client	The system should use Interactive Broker (IB)
Client	The system should trade using momentum algorithms
Client	The team should restrict the users of the system
Client	The system should use the Paper Trading mode of IB
Client	The system should provide user options of manual and automatic trading
Client	The system should use Q-Learning and Dyna
Client	The system should accurately predict stock performance
Client	The system should operate profitably

2) Constraints

While the objectives define what the project must do to meet the client's needs, constraints impose additional requirements on the project design. During the client interview, the design team documents any constraints the client imposes on the project. In addition, the design team reviews the client's objectives and constraints to identify any additional constraints necessary for the project. Table II lists the initial constraints captured in the initial client interview or subsequently determined by the design team.

TABLE II. CONSTRAINTS

Client Interview	
Source	***Constraints***
Client	The system must have minimal material costs
Design Team	The system must have trade simulator capabilities
Design Team	The system should have cross-platform capabilities

3) Assumptions

The final phase of the needs analysis consists of the design team building upon the objectives and constraints to identify any assumptions or limitations that further impact the project design. Assumptions are the engineering decisions made by the design team. Limitations result from decisions that impose design restrictions that are outside the design team's control. Limitations often result from physical laws or physical limits resulting from the technology choices. Finally, the design team document any assumptions or limitations that impact the project's design. Table III Assumptions lists the Assumptions for the Stock Market Trader Bot. Next, the design team identifies any limitations to the project's design.

TABLE III. ASSUMPTIONS

Needs Analysis	
Source	***Assumptions***
Design Team	The system only analyzes publicly traded stocks
Design Team	The system operates on the United States stock market
Design Team	The user is authorized to conduct requested transactions
Design Team	The user has installed IP Trader Workstation (TWS)
Design Team	The user has minimal knowledge of the stock market

4) Limitations

Table IV Limitations detail the project limitations identified during the needs analysis.

TABLE IV. LIMITATIONS

Needs Analysis	
Source	***Limitations***
Design Team	The user's TWS has the appropriate network connectivity
Design Team	The user's prior trading knowledge may impact the effectiveness of the BOT's performance
Design Team	The user's Interactive Broker account is active
Design Team	The lack of availability of historical stock data may limit BOT performance

5) Objectives

After compiling the objectives, constraints, assumptions, and limitations, the design team refines the objective lists. The team sorts the objectives into one of four categories: Safety, Progressive Training, Marketability, and User-Friendliness. Objectives categorized as safety identify goals related to minimizing the risk of harm to the system's users. Progressive training refers to implementing machine learning methodologies and applying artificial intelligence techniques to the system. Marketability impacts the commercial viability of the product. User-Friendliness enhances the user experience of the system. The team reviews the refined objectives with the client.

Table V presents the final project objectives, sorted by category, and assigned an objective ID number. Additional brainstorming sessions prospective meets to review the goals and objectives of the client.

TABLE V. PROJECT OBJECTIVES

Project Objectives			
Objective ID	***Source***	***Category***	***Limitations***
S1.1	Client	Safety	The BOT should be safe to use for novice and experienced traders
S1.1	Team	Safety	The BOT should minimize risk
S1.2	Team	Safety	The BOT should securely execute trades
S1.2	Team	Safety	The BOT should have cross-platform capabilities
P2.1	Team	Progressive Training	The BOT should successfully predict future performance
P2.1	Team	Progressive Training	The BOT should implement Machine Learning
P2.1	Team	Progressive Training	The BOT should analyze market behavior
M3.1	Client	Marketability	The BOT should be generally profitable
P2.2	Client	User-Friendly	The BOT should be easily upgraded
P2.2	Team	User-Friendly	The BOT should be fully modular
U4.1	Client	User-Friendly	The BOT should operate in both automatic and manual modes
U4.2	Client	User-Friendly	The user interface should be user-friendly
C1	Client	Constraint	The BOT should have no cost to build
F1.1	Client	Function	The BOT should interface with Interactive Brokers
F1.2	Client	Function	The BOT should provide an option for simulated trading
F1.3	Client	Function	The BOT should use momentum trading methods

However, Table V contains constraints and functions in addition to the categorized objectives. In this context, function indicates the suggested objective is a desired project use. Finally, we remove the functions and constraints from the table to produce the final version of the objectives table, Table VI Objectives.

TABLE VI. FINAL PROJECT OBJECTIVES

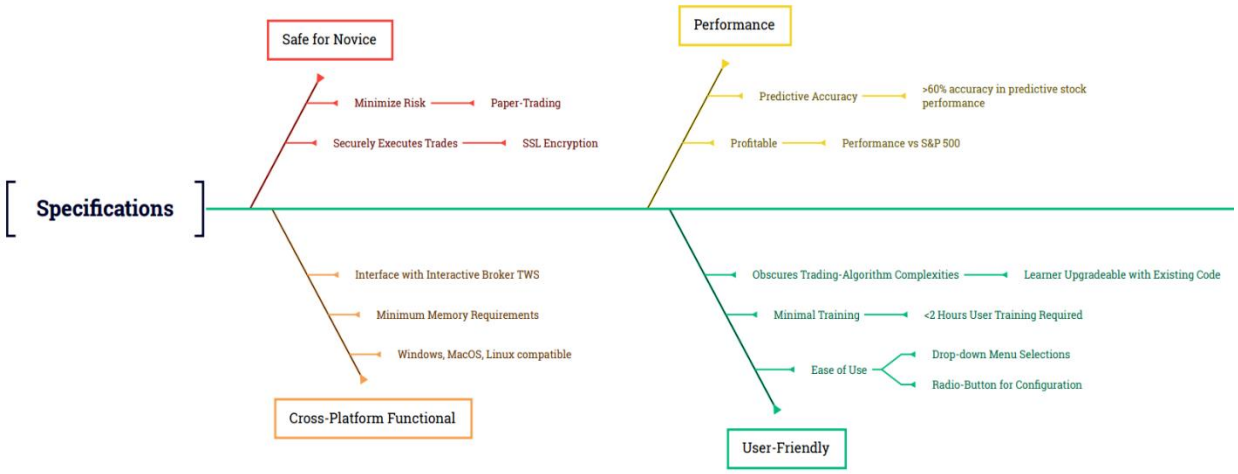
Project Objectives			
Objective ID	***Source***	***Category***	***Limitations***
S1.1	Client	Safety	The BOT should be safe to use for novice and experienced traders
S1.1	Team	Safety	The BOT should minimize risk
S1.2	Team	Safety	The BOT should securely execute trades
S1.2	Team	Safety	The BOT should have cross-platform capabilities
P2.1	Team	Progressive Training	The BOT should successfully predict future performance
P2.1	Team	Progressive Training	The BOT should implement Machine Learning
P2.1	Team	Progressive Training	The BOT should analyze market behavior
M3.1	Client	Marketability	The BOT should be generally profitable
P2.2	Client	User-Friendly	The BOT should be easily upgraded
P2.2	Team	User-Friendly	The BOT should be fully modular
U4.1	Client	User-Friendly	The BOT should operate in both automatic and manual modes
U4.2	Client	User-Friendly	The user interface should be user-friendly

With the project's objectives finalized, we present the objectives to the client for review and approval and progress to defining the project's specifications. In conclusion, this section is important because it shows priority towards the client's needs as well as the objectives outlined for the product. By outlining the objectives, assumptions, and limitations as done in this section, there is a clear correlation to the needs analysis of the project and provides a smooth transition to the need's specification section.

B. Needs Specification

The next step in building the proposal consists of creating the project specifications. The design team meets to construct the project specifications after the collaborative review of the Needs Analysis. The specifications provide the engineering requirements to deliver the project objectives. The specifications are technical measurements of how well the project meets the desired client's objectives, user's expectations, and the team's design goals. The following fishbone diagram, figure 1, serves as a simple representation as to how the needs specification is structured.

Fig. 1 Fishbone Diagram



Specifications originate from objectives, assumptions, limitations, or the operating environment. However, each specification is related to a specific Objective ID to ensure we have met or exceeded the design goals. Physical characteristics often drive the value of a specification for hardware and can include power requirements, environmental conditions, drop resistance, cable interfaces, and other details as necessary. For software systems such as the project, the project's specifications relate to the application's interface with the external systems. Software specifications often include compatible operating systems or communications protocols. Table VII provides the Stock Market Trader Bot specifications.

TABLE VII. SPECIFICATIONS

Specifications		
Objective ID	***Specification***	***Value***
S1.1	Secure transactions	Encapsulates transmitted data using Secure Socket Layer (SSL) encryption
S1.1	Safe Trading	Emergency Stop button to halt all trading activity
S1.2	Operating Systems	Compatible with Microsoft (Windows 10 or newer), macOS, and Linux operating systems
S1.2	Memory Requirements	Requires minimum 512 KB for installation, 128 KB RAM
S1.2	Accuracy	> 60% accuracy of predicted stock activity
S1.2	Ease of use	< 2 hours of learning required for new users and novice traders
U4.3	Profitable Trading	α >1, trading algorithm outperforms S&P 500

Industry standards drive most of the specifications in Table VII. SSL provides modern encryption of transmitted data while minimizing the lag in executing trades. Additionally, Interactive Broker's TWS uses SSL to communicate with third-party applications. An emergency stop feature provides additional protection for novice traders. The Emergency button is visible from all screens and pauses all trading activity. Compatibility with the three major modern operating systems ensures the portability of the application. Memory requirements are preliminary and subject to modification as the Project development matures.

The Bot's trading algorithm will try to accurately predict the future performance of specified stocks. This specification is not an indicator of profitability as the algorithm may also take actions to minimize losses. Trading profitability uses an alpha (α) index, which represents the ability to beat the market's average return. Finally, the system's intuitive user interface allows new users including novice traders, to begin executing trades within 2 hours of application training.

*C. **Feasibility Anaysis***

In this subsection we present the findings of our feasibility analysis. A feasibility analysis reviews several factors that may contribute to or negatively impact the successful completion of an engineering project. Our analysis includes technical, resource, schedule, economic, cultural, legal, and marketability reviews. We identify potential factors related to each of the previously mentioned feasibility topics and develop a feasibility score to predict the likelihood of completing our project successfully.

We will provide the feasibility score per topic which quantifies the project's likelihood of success. We also identify critical factors for completing our project successfully and on time. Finally, we combine the total feasibility assessments to calculate the project's weighted feasibility score which provides a general indication of the possibility of successful project implementation.

We begin with the feasibility assessment for each topic. Each assessment includes the topic, its key attributes, the team's score for the attribute, our justification, and a proposed solution for the attributes as necessary. These assessments include a score-based rating of our design team's ability to meet the design requirements of each factor. These scores may vary from 1 to 5, where 5 indicates the highest expectation of success and 1 indicates the most significant challenges to success. For each topic, we provide an overall score, the average of the key attributes. We use these scores to complete the project feasibility assessment, where a score above 3 indicates a more than likely successful project completion.

1) Technical Feasibility Assessment

The technical feasibility assessment identifies any technological requirements or limitations that could prevent the successful completion of our project. This assessment answers whether the technology exists to complete our design project successfully.

We reviewed the technology required to deliver our project, most notably asking the question of whether the technology presently exists to complete our project. As our project operates on existing systems and integrates into the published APIs for established software, we are confident in the availability of existing technology to complete the project. Additionally, we consider the availability of the required technology and software libraries for project completion. The technical feasibility score of 4.7 indicates the availability of the technology necessary to complete the project, as shown in Table VIII.

TABLEVIII. TECHNICAL FEASIBILITY ASSESSMENT

Technical Feasibility Assessment			
Attribute	***Score***	***Justification***	***Solution***
Does the required technology exist to complete the build the project?	5.0	Our project uses proven stable software and technologies, including the Python programming language and machine learning methodologies.	No Solution necessary
Is the necessary technology available for our use?	5.0	We will use the published Interactive Brokers Application Programming Interface (API) to implement available Python libraries for finance.	No Solution necessary
Will we need to design a new library?	4.0	Implementation of our algorithm will require the development of a project-specific library.	Ideally, we will use available open-source libraries to implement our project, only relying upon custom development if necessary.
Total Score	14.0		
Average	4.7		

2) Resource Feasibility Assessment

In our resource feasibility assessment, we ask whether the design team has adequate resources to complete the project. We evaluate the resources available on our team and the team's expertise level in the key areas. We consider Python programming with Machine Learning and Stock Market trading expertise necessary to develop a profitable trading algorithm. In this area, we identified possible solutions for overcoming the challenges, including obtaining additional training resources to address technical deficiencies. The resource assessment score of 3.3 indicates negligible resource risk mitigated given the availability of training resources, as shown in Table IX.

TABLE IX. RESOURCE FEASIBILITY ASSESSMENT

Resource Feasibility Assessment			
Attribute	***Score***	***Justification***	***Solution***
Do we have enough resources?	5.0	Our team of 4 Engineering students provides the necessary resources to complete the project.	No solution necessary
Do we have the python programming skills necessary to complete the project?	3.0	Not all team members have python programming experience, machine learning, and experience within the financial markets.	The team commits to complete advisor-suggested readings and online courses to improve the known knowledge deficiencies, including Python for Finance resources and access to the Interactive Brokers API guide.
Do we have the stock market expertise to complete the project?	2.0	The design team has limited experience with the stock market.	The team will conduct the necessary research and complete suggested online courses to gain a fundamental knowledge of the financial markets required to complete the project.
Total Score	10.0		
Average	3.3		

3) Schedule Feasibility Assessment

The schedule assessment asks if the team can complete the project on time. We consider the required project deliverables and the projected scheduling to confirm that the project's schedule is reasonable. Additionally, we measure the impact of the semester break and its potential for delaying the project delivery. The 4.0 assessment score reflects a high expectation of on-schedule project completion, shown in Table X.

TABLE X. SCHEDULE FEASIBILITY ASSESSMENT

Schedule Feasibility Assessment			
Attribute	***Score***	***Justification***	***Solution***
Can we complete the project on time?	5.0	The project schedule ensures the completion of deliverables for timely project completion.	No Solution necessary
Can we complete the preliminary design review on time?	4.0	Following the course outline, we will perform the preliminary design review on schedule.	The team meets regularly to review individual progress on assigned tasks and deliverables.
How can we overcome challenges related to semester break?	3.0	The project schedule includes a semester break between July 29th and August 22nd.	The design team will continue to meet throughout the break to review project deliverables and resources.
Total Score	12.0		
Average	4.0		

4) Economic Feasibility Assessment

The Economic Feasibility assessment measures the ability to complete the design project under budget. As we are building a software system, the design team does not anticipate hardware or software costs for our project. However, we consider additional expenses incurred to complete the project, such as online courses in Python programming. The assessment score of 4.7 indicates the low impact of economic factors affecting project delivery, as shown below in Table XI.

TABLE XI. ECONOMIC FEASIBILITY ASSESSMENT

Economic Feasibility Assessment			
Attribute	***Score***	***Justification***	***Solution***
Do we expect to incur any material costs in delivering the project?	5.0	No material costs of development	No Solution necessary
Can we cover the training costs to gain the necessary skills and fundamental knowledge?	4.0	Suggested online materials have minimal cost	The design team has indicated a willingness to cover the cost of any additional training cost incurred initially. Training costs will be included in development costs for the project and reimbursed if the project is externally funded.
Can we complete the project as budgeted?	5.0	The team expects no material costs and minimal training costs in addition to the labor costs of developing our project.	No Solution necessary
Total Score	14.0		
Average	4.7		

5) Cultural Feasibility Assessment

The cultural assessment identifies any social or cultural barriers to successful project completion. In our evaluation, we determined there are no significant cultural barriers as there are already several algorithmic trading systems commercially available to the public today. Additionally, we consider the potential social benefit of completing our project and any method we can take as a design team to limit any adverse cultural reactions to the project. The assessment score of 5.0 shown in Table XII indicates negligible cultural factors impacting project completion.

TABLE XII. CULTURAL FEASIBILITY ASSESSMENT

Cultural Feasibility Assessment			
Attribute	***Score***	***Justification***	***Solution***
Can we prevent any cultural barriers to project completion?	5.0	Numerous various algorithmic trading programs exist and are universally accepted.	No Solution necessary
Can the project provide any social benefits?	5.0	The project intends to provide reliable market performance for novice traders, potentially providing a safer method of introducing the market to marginalized communities.	No Solution necessary
Can we predict or prevent any social objections to the project?	5.0	The project can assist in democratizing the stock markets by providing a safer, potentially profitable entrance to the stock market.	No Solution necessary
Total Score	15.0		
Average	5.0		

6) Legal Feasibility Assessment

In completing our legal assessment, we consider any legal obstacles to completing the project. The design team anticipates significant legal requirements as the project is related to executing stock market trades. However, in our assessment, we note the requirements to adhere to all legal requirements and provide liability protection from any users of our system. Our legal assessment score of 3 indicates the awareness of significant legal challenges and the design team's intention to consider all legal requirements in our design.

TABLE XIII. LEGAL FEASIBILITY ASSESSMENT

Legal Feasibility Assessment			
Attribute	***Score***	***Justification***	***Solution***
Can we prevent any legal barriers to project completion?	3.0	Beyond the proprietary algorithm we are developing, our project builds upon open source and publicly available information.	No Solution necessary
Can we ensure the project adheres to any financial-related laws?	3.0	Our project intends to provide investment advice using our algorithm, so we must comply with all financial requirements.	The team will need to research and comply with all applicable laws related to providing market trading advice.
Can we minimize or eliminate any legal risk associated with providing our Trader Bot?	3.0	While initially designed for private use, the design team will need to account for any possible uses and potential outcomes.	The design team will need to provide explicit terms and conditions requiring the users of the Trader Bot to accept all associated risks with live market trading, including adhering to local and international laws.
Total Score	9.0		
Average	3.0		

7) Marketability Feasibility Assessment

The Marketability assessment considers the likeliness of market acceptance of the design project. Algorithmic trading systems are standard in today's market. The assessment score of 5.0 indicates the general market acceptance of algorithmic trading systems.

TABLE XIV. MARKET FEASIBILITY ASSESSMENT

Marketability Feasibility Assessment			
Attribute	***Score***	***Justification***	***Solution***
Will this project be generally accepted?	5.0	Algorithmic traders are common in the financial market.	No Solution necessary
Total Score	5.0		
Average	5.0		

8) Weighted Feasibility Ranking

Table XV is the feasibility ranking chart. The weighted ranking of each topic reflects the relative importance of the topic in each column as it relates to the topic given in the corresponding row. Scores indicate the relative importance between topics: 1 - equally important topics, 3 - slightly more important, 5 - moderately more important, 7 - strongly more important, and 9 - extremely more important. Conversely, scores less than 1 indicate less importance than the comparison topic. These values are the inverse of the previous values. Accordingly, relational scores values indicate that the topic in the vertical topic column is less important than the topic in the horizontal header row: 0.11 - extremely less important, 0.14 - strongly less important, 0.20 - moderately less important, and 0.33 - slightly less important.

We also calculate the Geometric Mean for each topic score and compute the weighted rankings of the topic. (1) defines the Geometric Mean

$$G._{MEAN} = (A_1 \cdot A_2 \cdots A_N)^{(1/N)} \quad (1)$$

We calculate each topic's weight (w) ranking by dividing the topic's Geometric Mean by the sum of all the Geometric means. According to the results of the weighting, we see the technical and legal factors are the most important and are equally important. Likewise, as shown in Table XV, the cultural and economic factors are expected to be the least important factors.

TABLE XV. FEASIBILITY WEIGHTING

Feasibility Weighting									
	Technical	***Resource***	***Schedule***	***Economic***	***Cultural***	***Legal***	***Marketing***	***G. Mean***	***w***
Technical	1.00	3.00	3.00	9.00	9.00	1.00	9.00	3.51	0.32
Resource	0.33	1.00	3.00	7.00	7.00	0.33	3.00	1.74	0.16
Schedule	0.33	0.33	1.00	5.00	5.00	0.33	3.00	1.16	0.10
Economic	0.11	0.14	0.20	1.00	1.00	0.11	0.20	0.26	0.02
Cultural	0.11	0.14	0.20	1.00	1.00	0.11	0.20	0.26	0.02
Legal	1.00	3.00	3.00	9.00	9.00	1.00	9.00	3.51	0.32
Marketing	0.11	0.33	0.33	5.00	5.00	0.11	1.00	0.62	0.06
							Total	11.05	1.00

9) Weighted Feasibility Scores

Finally, we determine the weighted feasibility score of the project. This score represents the likeliness of project completion, given the current project conditions. Projects scored three or higher are deemed appropriate for completion. For each topic, the weighted score is the feasibility assessment score multiplied by the topic weight. The project's feasibility score is the sum of the weighted scores of the topics divided by the sum of the topic weights, given in (1) below. Table XVI shows our project feasibility score of 3.87, indicating a high likelihood of successful project completion.

$$\textit{Weighted Average} = (\Sigma \textit{ Weighted Score})/ (\Sigma \textit{ Weight}) \quad (1)$$

TABLE XVI. WEIGHTED FEASIBILITY SCORE

Weighted Feasibility Score			
Feasibility Type	***Score***	***Weight (w)***	***Weighted Score***
Technical	4.70	0.32	1.50
Resource	3.30	0.16	0.53
Economic	4.00	0.10	0.40
Schedule	4.00	0.02	0.08
Cultural	5.00	0.02	0.10
Legal	3.00	0.32	0.96
Marketing	5.00	0.06	0.30
Totals	29.00	1.00	3.87
Weighted Average (Feasibility Score)			3.87

In this subsection, we have detailed our process for conducting a feasibility analysis of our project. We included assessments for each topic that may impact our ability to complete the project as intended and on time. Our feasibility score of 3.87 indicates a high likelihood of success in completing our project successfully and on time, taking the various feasibility types into consideration. In the following section, we will discuss our risk management process and mitigating actions we plan on taking to ensure we manage any risk to the project delivery.

D. Marketability

The Marketability Analysis of a product is imperative as it provides a great wealth of information pertaining to how well the product will do in a consumer's market. With no actual market to identify for a product, its success will be hindered. Identifying competition, similar projects, appealing to relevant corporations, and targeting specific customers are all factors that attribute to the success of the product. To ensure the success of the stock trading bot, the team has compared its own data and research with similar projects on Kickstarter that were able to gain funding followed by implementation. By issuing a comparison between two similar projects related to the team project, the ascertaining of information about successful products provides a clear picture to which consumers are of interest. In particular, targeting projects that are useful for hedge funds and employ strategies such as shorting, longing, and arbitrage are essential.

A strategy called shorting occurs when a trader borrows shares from a broker and sells them when they expect the stock price to fall. If successful, the trader can then buy the shares back at a lower price and return them to the broker. Longing refers to maintaining a secure position with a stock the trader expects to rise in value in the future. If successful, the stock can be sold later for a profit since the price has risen in value.

To briefly introduce the concept of arbitrage, lets declare a general definition as to what is defined as arbitrage in our team. Arbitrage refers to taking advantage of the price of a single asset when it dips even slightly when compared to its original price. An example of arbitrage would be geographic arbitrage, where two collocated hedge funds are a distance apart. One hedge fund may read a price of a certain stock at a slightly higher or lower price than what appears at the second hedge fund due to the geographic difference. If this occurs, one hedge fund may attempt to buy the lower stock in the first hedge fund and sell it to the second hedge fund with a higher price even if the difference is almost minute.

The first product we will be discussing is the Zeke. This section will be outlined by describing the project summary, fundraising strategies, technological overview, and system description. To conclude, we will tie in how this product correlates to our design and implementation phases of the product.

1) Zeke

Zeke, The, "Automated stock market trading software to generate wealth" [21] Was developed in Savannah, Georgia by Maverick Business Solutions and launched in July 2019. Maverick Business Solutions (MBS) is a trading platform technology company focused on software development to maximize profitability in trading within the stock/currency market.

a) Project Summary

Zeke is a custom designed software program that evaluates the Stock/Currency market to find optimal opportunities to maximize potential profit. "Zeke does not become fatigued, confused, or emotional and serves as a surrogate to monitor hundreds of strategies set for the public market using a cultivation of technical and fundamentals" [21]. Once the program is started it can operate fully unattended even if the computer is off. What makes Zeke unique, is that Zeke is capable of executing user-defined trading strategies continuously and instantaneously without user intervention.

Fig. 2. Zeke Mobile App [21]

b) Fundraising Strategy (Rewards)

Zeke was able to raise USD$10,299 from 20 backers within its timeframe. The fundraising rewards that were implemented consist of virtual and physical thank you notes including small tokens such as a pen or bumper sticker. The reward system set by Maverick Business Solutions is displayed below:

- At USD$1 or more: Virtual High five + Frequent project updates
- At USD$11 or more: A thank you card signed by partners and a fountain pen with the MBS logo on it with a color of your choice.
- At USD$15 or more: You will receive everything in the $11.00 package plus a bumper sticker with the logo on it, as well as a cutting board.

c) Technology Overview

The Zeke software contains a simulation mode that allows beginners to use Zeke to learn how the stock/currency market works and how to have an effective strategy in the long run. MBS Academy is a suite of training tools that subscribers will receive to learn trading strategies from the basics to an expert level. The software itself is compatible and will automatically trade with some of the most popular brokers such as TD Ameritrade, Forex.com, and Interactive Brokers. Slyness mode is an option which prevents hedge fund managers, competitive brokers, and rival investors from viewing your order limits and manipulating the market to stop loss you into a loss. This mode hides the price until the order is executed as per the user's set requirements and conditions.

TABLE XVII. Zeke SPECIFICATIONS

Component	Feature	Functionality
Past data archives	Back Testing	To let a user run a simulation to see how the system would perform during a past market pattern
Zeke App	Slyness Mode	Allows a user privacy from hedge fund managers, brokers, and rival investors
Trading Strategy	Mirror Trading	Mimics trades executed by seasoned and experienced forex traders in an efficient manner

d) System Description

Zeke software system is fully automated, and trigger based. The capacity for trading is up to 1500 trades per week and is user friendly for anyone who has not traded before. The automation feature maximizes portfolio growth while allowing the user to set the triggers and walk away. Back

testing provides an archive of past data to test against while allowing the user to see a simulation showing how the system would theoretically perform if it had been active in a past market environment. This allows the use of practice accounts to mitigate risk using real money.

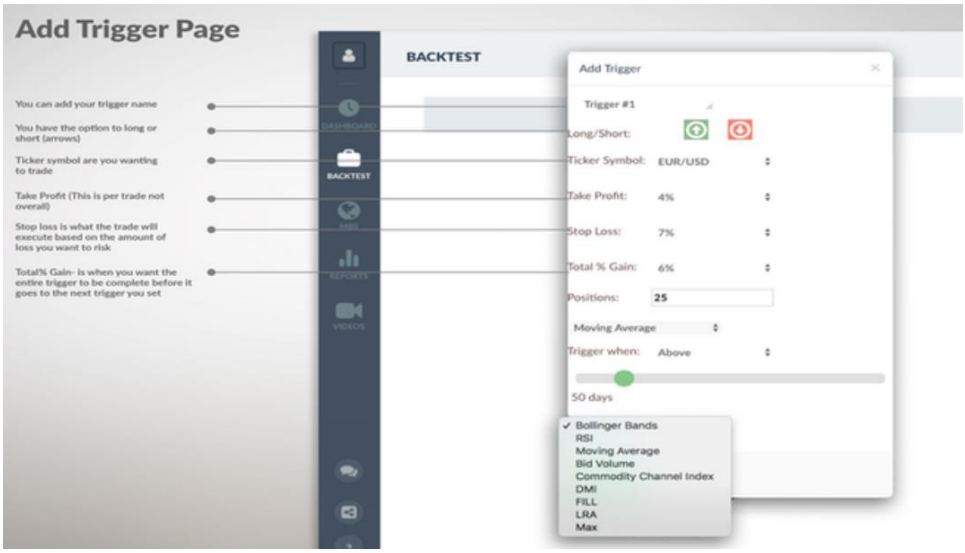


Fig. 3 Zeke Trigger Page UI [21]

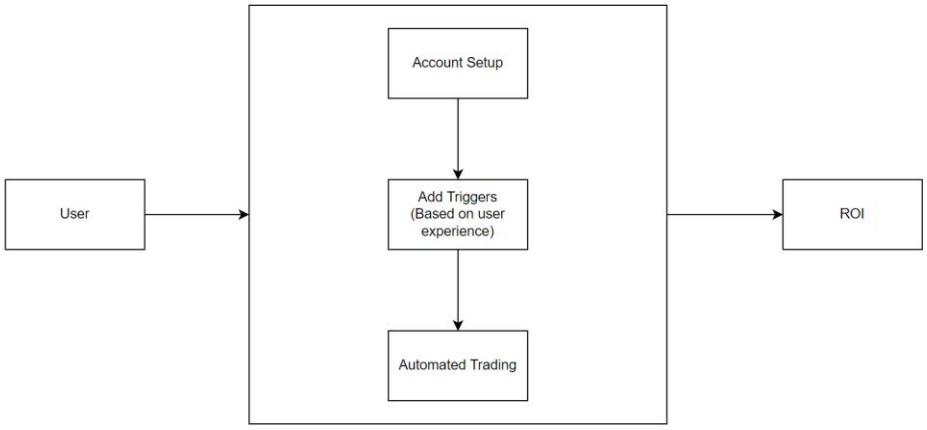


Fig. 4 Zeke Functionality Block Diagram [Team Made]

2) Finnhub – The Free Stock API for Investors

Finnhub is a free stock API (Application programming interface) to access real-time market data and institutional-grade financial data to assist in investments. Developers and retail traders are the main users of Finnhub, providing them with "financial data at an affordable price without having to inquire through complex licenses from data providers". [19]

a) Project Summary

Finnhub's main function is to democratize financial data while making use of "state-of-the-art machine learning algorithms to collect, clean, and standardize data across global markets" [19]. Real-time stock prices for over 60 stock exchanges around the world are provided. Also included are real-time market news and news sentiment analysis, estimate upgrade/downgrade decisions, earnings calculator, IPO calendar, and ICO calendar are a few features of the API. The project was launched on Kickstarter on January 24, 2020, with a goal of $500 to provide their information for free to developers. The goal was successfully met a little over 2 weeks later.

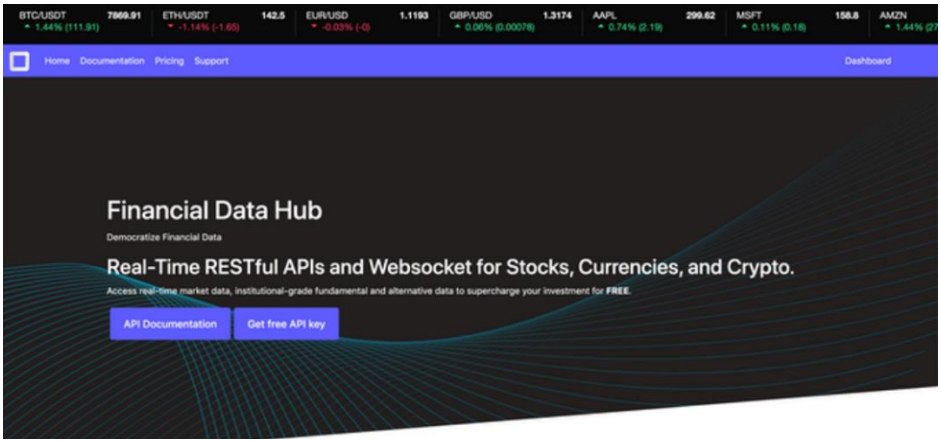


Fig. 5. Finnhub's Home Page [17]

b) Fundraising Strategy (Rewards)

- Pledge $50 or more, get 1 month of basic plan. Included with 150 API calls per minute
- Pledge $100 or more, get the standard plan. Included with 300 API calls per minute
- Pledge $150 or more, get the professional plan. Included with 700 API calls per minute

c) Technology Overview

Open-source libraries for the Finnhub API includes languages such as Python, Go, JavaScript, Ruby, Kotlin and PHP. Datasets that are accessible and free to download include SEC financials, SEC fillings metadata, and S&P 500 futures tick data. From the REST API, data that can be returned consists of income statements, balance sheets, cashflow statements, and earning call statements.

TABLE XVIII. Finnhub Feature, Purpose, and Functionality

Feature	Purpose	Functionality
REST API	Data accessibility	To allow a user to retrieve specified data from Finnhub through HTTP endpoint requests
Open Estimate	Gathers voluntary data	Collects data from the Finnhub community provided by the community itself to calculate estimates on stock growth
Open Data	Open datasets	Provides students, researchers, and investors with data to test their own algorithms and predictions

d) System Description

The API is centered around REST (Representational state transfer) principals which return JSON-encoded responses via standard HTTP response codes. Limitations include a "30 API call per second and will return a status response 429" [17] if the limit is exceeded. In the testing sandbox, the environment has a 60 calls per minute limitation. Categories of data that can be accessed consist of stock fundamentals, stock estimates, stock price, ETFs and indices, mutual funds, bonds, FOREX, crypto, technical analysis, and alternative data. This data is sourced directly from exchanges such as ASX, ActivFinancial, EDI, and QuoteMedia. The data is then made available using SEC filings endpoint and International Filings endpoint while ETFs data are sourced from issuers' filing and public websites.

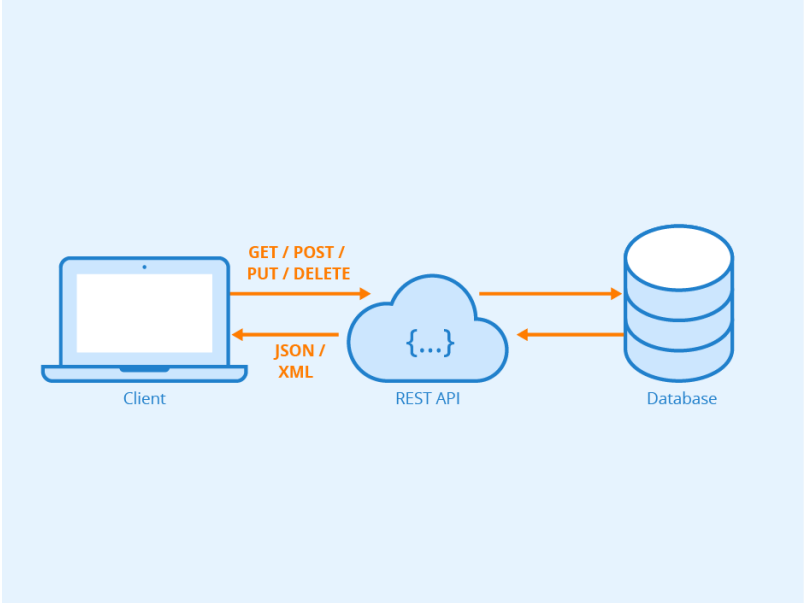


Fig. 6. REST API Architecture

After careful deliberation, similarities are drawn from the two products mentioned when compared to the design of the team's group project as well. Both products aim to use a technology that provides investment options and utilizes financial data for profit. This means that through these fundraising methods each campaign could be used and is applicable. Knowledge of these products and how they were funded will help the team think carefully about what demographics are being targeted. Zeke was notably more relevant especially with its usage of automated trading within the stock market. The fundraising success of Finnhub will be considered when the team will be considering funding options.

V. Risk Analysis

Every engineering project has the possibility of loss, defined as the project's risk. This subsection describes the methods we will take to identify, classify, and mitigate the risk specific to our project. Risk assessment includes identifying the risk, setting clearly defined risk thresholds, and organizing the risk based on the potential impact on the project. Utilizing the feasibility types from the previous subsection, we conducted brainstorming activities to identify possible risk factors.

We classify each identified risk factor according to the likelihood of occurrence and the severity of possible outcomes or losses should the risk become realized. Risk occurrences are classified as either Unlikely, Possible, or Very Likely. We rank the outcome severity in risk classes I, II, III, and IV. Class I risk are the lowest severity risks, with generally acceptable risks that do not require active risk management. Class II risk is the lowest risk class that requires active risk monitoring and can negatively impact the project. A Class III risk has a moderate impact, requires proactive risk management, exceeds the project's acceptable risk threshold, and could prevent project completion. Class IV risk offers the project's most significant risk and dramatically exceeds the acceptable risk threshold. Class IV risk requires urgent and immediate actions to prevent project failure.

A Risk Exposure Matrix consolidates each risk factor's classification as Low, Moderate, Severe, and Catastrophic. The risk matrix details the necessary level of proactive risk management actions. Proactive risk management actions include identification, classification, risk mitigation, risk monitoring, and correcting any deviation from the risk management plans.

A. *Risk Identification*

To identify possible risk factors, the team conducted a brainstorming session. The brainstorming session concluded in the fishbone diagram in Figure 7. To complete the fishbone diagram, we identify possible risk factors that would prevent the completion of our project. Each risk is classified according to its corresponding feasibility topic.

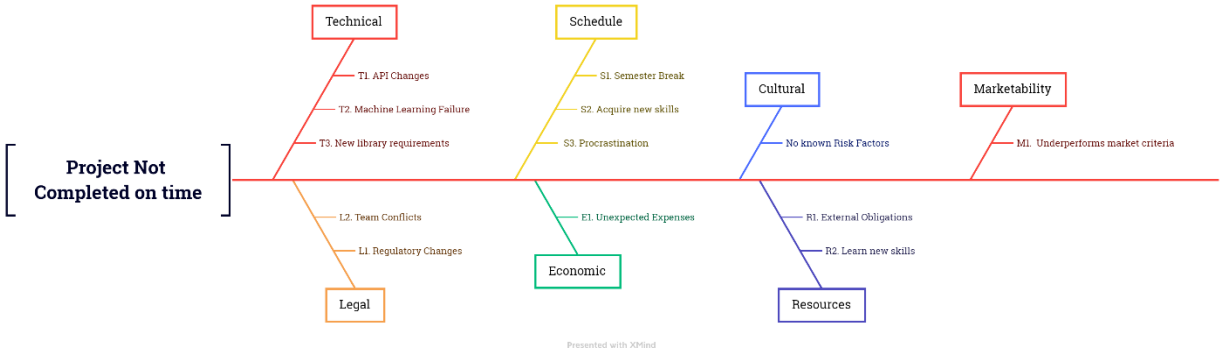


Fig. 7. Fishbone Diagram of Risk Factors

The resulting list of risk factors identified by the design team that could have a negative impact for project delivery is provided below with justification.

1) Technical

T1. Application Programming Interface (API) Changes – Any API changes made during our development could lead to unexpected delays in order to adjust our code to match the updated API.

T2. Machine Learning Failure – Machine Learning is a key component of our algorithmic trading system and meeting the project's performance specifications.

T3. The project requires New Library development – We will rely on code reuse to implement our algorithms and meet the project deadline. Any requirement to development extensive customized software libraries will have a negative impact on our delivery schedule.

2) Resource

R1. External Obligations impact the availability of team members – Our schedule is based upon the agreed upon availability of team members. However, multiple team members have full-time employment that may present occasional unavailability for our regularly scheduled team meetings.

R2. The requirement to learn new skills within the project schedule – Some team members are learning stock market trading strategies and python programming to contribute to the development effort. This could impact completing milestones on schedule if training is not completed as agreed by all team members.

3) Economic

E1. Unexpected development cost – The development is expected to have minimal material costs. Excessive unexpected cost would require the team to address possible funding for materials.

4) Schedule

S1. Semester Break during the project – The academic calendar includes an extensive break between the Summer and Fall semesters. During this period, team members will need to remain active and engaged to prevent unexpected delays in completing scheduled activities.

S2. Time required to acquire new skills – Design team members may need to complete available online python training with a focus on machine learning in addition to the project deliverables.

S3. Procrastination by the design team – While unexpected, procrastination can have detrimental impacts on team success, leading to missed deadlines or quality deficiencies.

5) Cultural

No known risk factors

6) Legal

L1. Regulatory Changes – Due to the financial aspect of the project, any regulatory applicable to algorithmic trading could adversely impact the requirements of the project and delivery schedule.

L2. Team Contractual Conflicts – Internal team conflicts can have a negative impact on the effectiveness and efficiency of the team. The conflict must be corrected in a respectful manner to maintain team cohesion.

7) Marketability

M1. Underperformance related to T2 – Any failure to meet the performance specifications would negatively impact market acceptance of the project.

B. Risk Matrix

Table XIX shows the risk matrix for our project. The risk matrix identifies the items requiring proactive risk management based upon the combination of likelihood and impact. From the risk matrix, we confirmed there are no catastrophic risk factors. The severe risk factors S1, R2, T2, and M1 will require proactive risk monitoring and

TABLE XIX. RISK MATRIX

Risk Matrix				
	Very Likely	Possible	Unlikely	Legend
Class IV				Catastrophic
Class III		T2, M1	S3	Severe
Class II	S1, R2	T1, T3, S2	E1	Moderate
Class I	S1, R1	L1	L2	Low

C. Risk Mitigation

Table XX shows the risk mitigation actions of our project. These actions provide proactive risk management to respond to any possible losses promptly.

TABLE XX. RISK MITIGATION ACTIONS

Risk Mitigation	
Task	***Actions***
T2, M1	Active measurement of Machine Learner performance
S1	Team members maintain regular communication of availability of any possible project slippage
R2	Team members will continue skill training during the semester break
T1	The team designates a member responsible for API functionality
T3	The team designates a member responsible for library management
S2	The team assigned and each member has committed to continuing our skills training during semester breaks
L1, L2, E1, S3	The team does not monitor Low-impact tasks but remains aware of their potential impact on the project

This subsection reviewed the risk analysis and mitigation activities necessary for continuous risk management. These actions are essential to ensuring design projects stay on track for successful delivery. While our project includes a severe risk related to the performance of our machine learning algorithm, we are confident in our ability to proactively monitor and mitigate the risk. Accordingly, the project falls within an acceptable risk level. In the next subsection, we will review the globalization characteristics of our project, including adherence to applicable local and global standards.

VI. Operating Environment

The Stock Market Trader Bot is specifically designed to operate in a predetermined environment. This section discusses our bot's necessary memory required to operate along with the interfaces we will use such as Trader Workstation. The design of systems must account for the different environments in which they may operate. Accordingly, the operating environment section outlines the specific conditions suitable for the system's operation. For physical systems, the environment description includes the limits of operational conditions, including input power requirements, temperature limits, moisture or dust tolerance, and many other possible environmental conditions. In software or virtual environments, the operating environment defines the virtual requirements necessary to install and deploy the software, such as operating system and storage requirements. In the remainder of this section, we will detail the requirements for the system.

The Stock Market Trader Bot is an algorithmic trading application. The system has been written using the Python programming language to provide flexibility in the operating environment. Theoretically speaking, this decision allows for system deployment on various modern operating systems, including Linux, Macintosh, or Windows. The application interfaces directly with the Interactive Broker Trader Workstation (TWS) using their published Application Program Interfaces (API). The BOT requires a network-connected device with an existing connection to an Interactive Broker Trader Workstation. The installation requires a minimum of **512 Kilobytes (KB)**, while operation requires a minimum of **128 Kilobytes (KB)** of available RAM.

In this section, we have detailed the expected operating environment for the application. These specifications allow a potential purchaser to evaluate the system's applicability to their environment. Section VII discusses the application's Intended Use(s) and Intended User(s).

VII. INTENDED USER(S) & INTENDED USE(S)

An imperative and highly significant aspect with respect to the sophisticated production and implementation of engineering products is clarified, concise, and intricate specifications of how the product will be used and by who it will be utilized by. To produce and propel effective, intelligent decisions relevant to the who/how questions, a comprehensible and transparent list of intended user(s) and use(s) provides effective consultation and orchestration to the project in its entirety, providing designing engineers a clearer picture as to the respective end-product goals.

A. Intended User(s)

Contemporarily speaking, the product is intended for a reduced number of individuals. More specifically, the focus is currently to have an utmost of ten total users. Furthermore, these limited number of users should be able to comfortably operate the Bot with minimal stock market exchange experience. The main target for current user(s) is:

- Traders of all experience: novice, intermediate, analysts, brokers, etc.

Furthermore, if the product and algorithm is re-equipped to not only involve more high-level statistical and financial predictions, but to evidently produce enhanced maximization of profits within a given time frame, then it may be considered to advance as commercially purchasable. In this case of algorithmic improvement, stake holding engineers will reevaluate this current state of intended user(s) and make an engineeringly-sound decision on if it should be publishable to the public or commercially/personally buyable. Amongst the diminished quantity of users, the four stakeholders, along with the aimed client, are ensured to possess the ability to utilize the product for personal use.

B. Intended Use(s)

The Bot's goal is to maximize the profitability of the user's portfolio while simultaneously minimizing the user's risk. The system uses a proprietary algorithm based upon proven financial models and machine learning modifications. The algorithm analyzes selected stocks' available historical market data provided by Interactive Broker (IB). The analyzed data includes current stock price, open price, close price, highest price, lowest price, and Volume within a given period. The application defines an ideal portfolio based upon the user's indicated risk criteria, portfolio details, and available funds and assets. We have designed the application for use in momentum trading optimized for portfolio management.

The intended use(s) of the Stock Market Trader Bot are:

- Analyze past performance of publicly traded stocks using an existing IB account.
- Predict future stock performance to accurately provide buy or sell suggestions
- Execute approved trade transactions to optimize the user's portfolio.

In conclusion, the application uses real-time or near real-time market data to drive the user's portfolio towards the suggested ideal portfolio. The system can execute trades on behalf of the user on their existing account, according to the user-defined trading method, manual or automatic trade operation. While in automatic operation, the user authorizes the Bot to execute trades that meet the user-specified criteria, including risk mitigation, profitability, and overall portfolio management targets. Manual trade operation requires user approval of trade execution for suggested trades that meet the execution criteria, which is the same as in automatic mode. The user's execute decision will be logged and recorded in manual mode.

VIII. BACKGROUND

At the core of complex and thoroughly executed engineering projects is an undermined, yet crucial part of the designing process: project background. The Stock Market Trader Bot requires work stemming from a variety of different, multiplexed specialties whether it be algorithmic/software development, mathematical computations, statistical analysis, etc. More specifically, the primary composition of mathematical computations included in this section is focused on the variety of different formulas and components involved in Hedge Fund Computing through active applications of machine learning and financial principles. Circling back to the different theories that played a role in the design phase, this background section will take an in-depth research-based approach into the Bollinger Bands, Grinold's Fundamental Law, Capital Asset Pricing Model, and Arbitrage Pricing Theory and provided readers a concise, yet effectively clear understanding of how these Nobel-prize winning theories came to motivate and affect the model. Lastly and arguably the most essential, is the software-focused, technologically sound, and systematically substantial implementation of the algorithmic trading behind Interactive Brokers.

This section is of heightened importance because it requires extensive research to appropriately provide readers and participating engineers a clear context as to not only the related projects currently in market, but what theories, approaches, and conceptual analysis was implemented and utilized by these analogous products. By providing this context, the client will be depicted a story as to how the evolution of the different aspects of the Stock Market Trader Bot came to be. More specifically, a distinct connection will be formulated as to how the system built upon these theories and products, advancing, and developing them to the next level.

A. Hedge Fund and Machine Learning Computings

The Stock Market is managed using a multitudinous number of varying types of exchanges. However, the types of funds, or pooled investments, can be separated into three dominant types: Mutual Funds, Exchange-Traded Funds, and Hedge Funds. A Mutual Fund is a type of financial vehicle made up of a puddle of money collected from many investors to invest in securities like stocks, bonds, and other assets [3]. Exchange-Traded Funds is a fund that offers investors an interest in a professionally managed, diversified portfolio of investments. But unlike Mutual Funds, Exchange-Traded Funds shares trade like stocks on stock exchanges and can be bought or sold throughout the trading day at fluctuating prices [4]. Lastly, Hedge Funds are funds that trade in relatively liquid assets and is able to make extensive use of more complex trading, portfolio construction, and risk management techniques in an attempt to improve performance, such as short selling, leverage, and derivatives. This section of the background will be focused on the impact and importance of Hedge Funds computing and its machine learning algorithms on the development of the Stock Market Trader Bot.

1) Summary

Hedge Funds are among the most computationally demanding environments with respect to the intricacy needed in promoting accurate decision-making and overall trading success. These Hedge Funds have immense infrastructure requirements such as vast databases, significant network connectivity, low latency, high bandwidth connectivity, real-time processing, and so on. At the heart of hedge funds is a deep, multidisciplinary, and multifunctional core with an emphasis on containing high level computing and computational capabilities. Moreover, Hedge Funds do contain a variety of variables and tangibles that come into play when aiming for profitable

financial success. The best way to provide a consistent and simple-to-follow layout as to how computing and machine learning plays a factor in Hedge Funds' approaches is to outline these different variables and their roles both concisely and accurately. Such variables include the Market, live/target portfolios, historical price data, trading algorithms, N-Day forecasts, etc. Moreover, the technological overview section will focus on going into further detail of each of these variables and how their contributions are directly correlated to the trading decisions made by these hedge fund managers

2) Technological Overview

The most effective approach to describing Hedge Fund computations is to work backwards from the start point which in this case is the Stock Market itself. This technological overview section will describe the main characteristics of the components and technologies used in Hedge Fund computations along with important information needed to understand the overall approaches of these mathematically demanding funds.

Referring to the stock market, two key components that have a correlation to the market is not only a trader's current, or "live" portfolio, but also the target portfolio, or where the trader aims and shoots for all their stocks to be. Within a portfolio, a specific quantity of stocks and options that were deliberately and thoughtfully selected for monetary investments by the trader or fund manager. These portfolios can range between one to hundreds, maybe thousands, of different stocks. To reach this target portfolio, traders must be active in the stock exchanges and operatively work with is called buy/sell orders. Moving on to the next component, Hedge Fund owners, along with experienced stock market traders, tend to formulate strong conclusions and financial decisions based off a stock's historical price data. Historical Price Data is used by investors and analysts to back-test pricing models, or investment strategies, to detect technical indicators or patterns of a certain stock [6]. Furthermore, off statistical analysis performed on the historical price data, machine learning plays a crucial role in the formulation of an effective trading algorithm that has an ultimate, primary goal to have one's "live" portfolio reach its "target" portfolio. A programming language, such as Python, allows for programmers to train a machine learning algorithm, or model, to analyze historical data. This sophisticated trading algorithm, along with a base target portfolio, access to historical price data, and the ability to buy/sell orders, allows for successfully formed hedge funds computations

It is undeniable that the statistical and mathematical analysis that is performed behind hedge funds computations is multiplexed and multivariable. Each of these mentioned components has distinct characteristics and relevant technology behind it. In the following System Description section, helpful diagrams will be provided to describe how each of these components works together. Furthermore, a more in-depth analysis of the interconnections evident not only within the components expressed in this section, but behind each one will be depicted using images and concisely written explanations to provide a succinct overview of hedge fund managers perform computations and exchanges in the market.

3) System Description

To provide a clear trackway relevant to a system's design, a detailed explanation of the interrelated parts must be accurately depicted using a collage of images and descriptions. In the following image, figure 8, is a diagram representing the top layer of the variables and components behind Hedge Fund Computations.

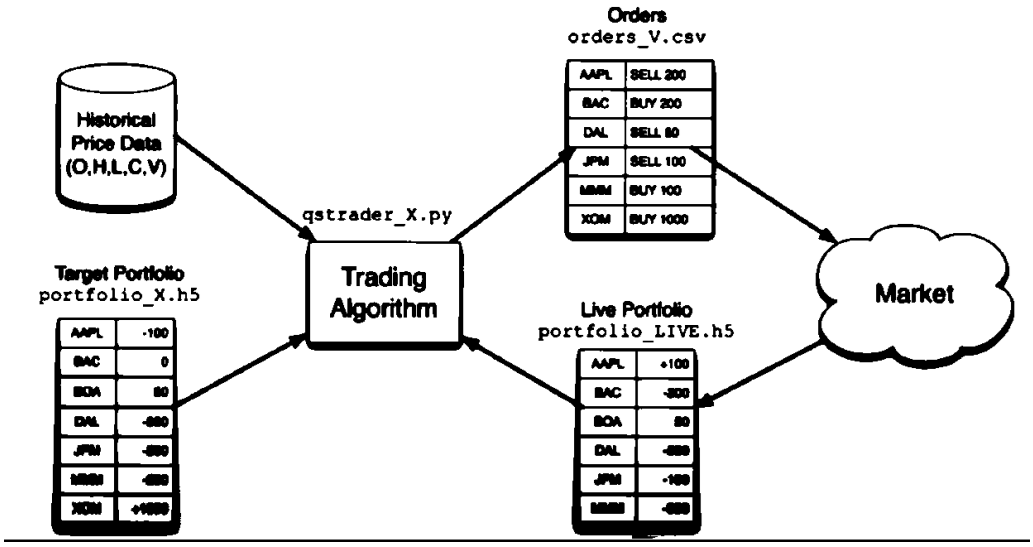


Fig. 8 The Computing Inside a Hedge Fund [7]

As portrayed in figure 8, we see there is direct correlation between the market and a trader's live portfolio. This is a result of fluctuating prices in the current market and how it actively changes the value of stock present in a given portfolio. This live portfolio is then seen to be handled by the trading algorithm which is the center piece of this contemporary diagram. The Trading algorithm is not only being fed a live portfolio, but it is also being given a target portfolio and historical price data. As mentioned in the technological overview section of this Hedge Fund computing concept, this trading algorithm uses the target portfolio along with data analytics of the historical prices to send buy and sell orders to the market. Overall, the end-goal is to effectively feed parameters and inputs into the algorithm to produce a target portfolio. In the follow diagram, figure 9, we will go further in-depth as to what other components ties into the Target Portfolio.

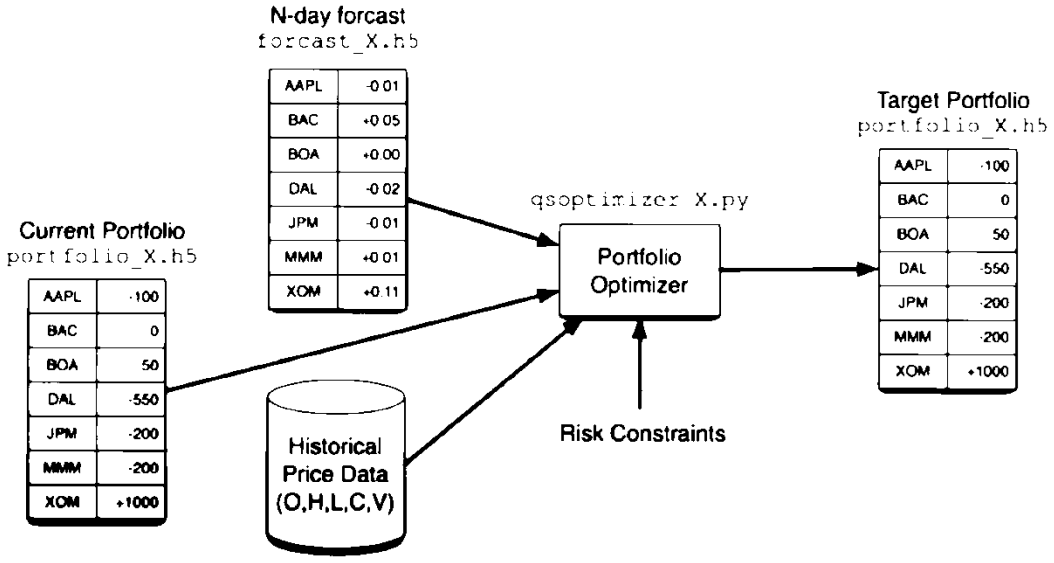


Fig. 9 The Interconnections of a Hedge Fund [7]

Figure 9 illustrates how a portfolio optimizer is used to form a trader's target portfolio. More specifically, to appropriately optimize a portfolio, components such as Historical Price Data, the current portfolio, and the N-Day forecast have an impact on its conclusion. The current portfolio is significant in its contributions because it provides a more faultless emphasis on which specific stocks should possess a higher focus in order to optimize the portfolio in its entirety. With respect to the N-Day Forecast that is directly related to the Portfolio Optimizer core, machine learning models and algorithms are trained using financial theories such as Bollinger Bands and the Arbitrage Pricing Theory to provide an accurate attempt at predicting the price of a given portfolio stock; thus, overall, optimizing the portfolio.

For figure 10, once again, a more in-depth approach is provided as to how machine learning is used to apply a valid prediction of a stock using N-Day forecasts. More specifically, machine learning is applied to a forecasting algorithm that is also fed Historical Price Data and Information. This information is relevant because company news is proven to spark reaction with investors known as price discovery. Furthermore, investors will process new information and decide how to approach their buy/sell orders with respect to the pertaining company as a result of new information [8]. Using principles of Artificial Intelligence and Deep Neural Networks, an algorithm can simulate and predict reactions to company news and anticipate how the stock will react. The different parts in Hedge Funds computation whether it be the forecasting algorithm, along with the Trading algorithm that connects the N-Day forecast and Target Portfolio, attempts to construct an effective plan to provide Hedge Fund managers a profitable experience in the market.

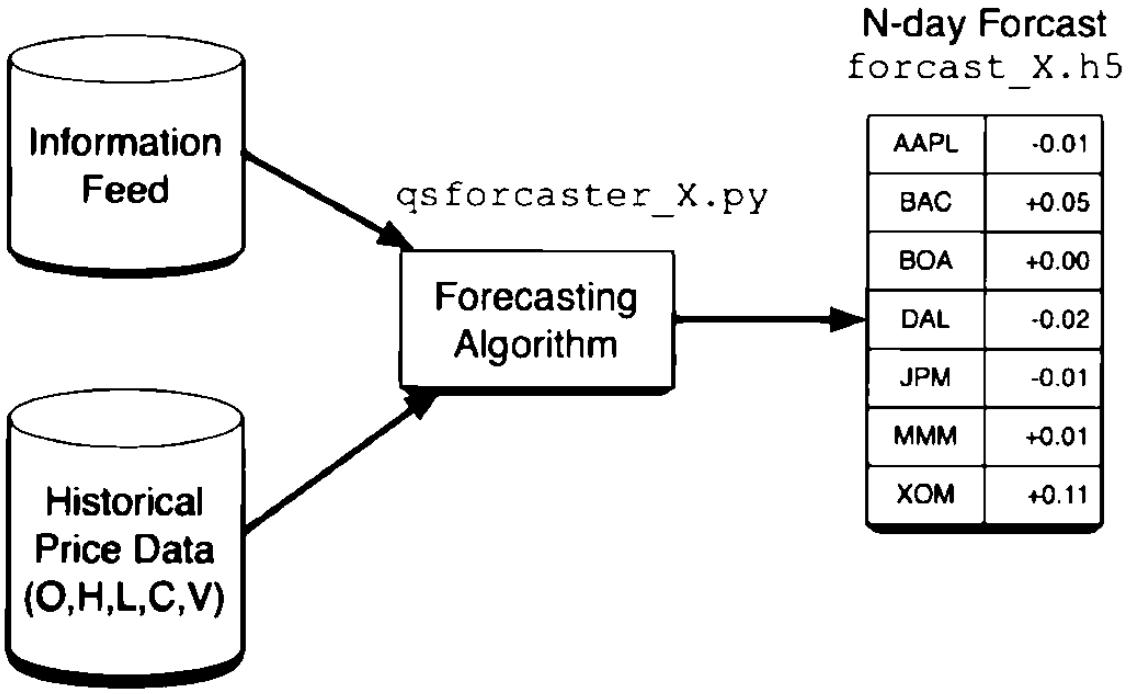


Fig. 10 The Computing Inside N-Day Forecasts [7]

B. Bollinger Bands, Grinold's Fundamental Law, Capital Asset Pricing Model, and Arbitrage Pricing Theory

1) Summary

Several different theories have made an undeniably strong impact in the way investors approach the market. However, among the most impacting and most correlating to the projected system, is the Bollinger Bands (BB), Grinold's Fundamental Law (GFL), Capital Asset Pricing Model (CAPM), and Arbitrage Pricing Theory (APT). The BB are known specifically for their ability to help determine whether prices are high or low on a relative basis through the use of both upper and lower bands in conjunction with a moving average [9]. The GFL is widely used to validate adding securities to an optimization universe, adding factors to forecast return, trading frequently, and reducing constraints [10]. The CAPM is aimed at describing the relationship between systematic risk and expected return for assets, particularly stocks [11]. Lastly, the APT was built upon its predecessor, CAPM, and has a primary purpose of trying to pinpoint the fair market price of a security that has the possibility of being temporarily incorrectly priced, indicating a buy/sell opportunity is evident. Following these summaries of each theory, the technological overview section will focus on providing specifics on each component and variable that contributes to the different theories.

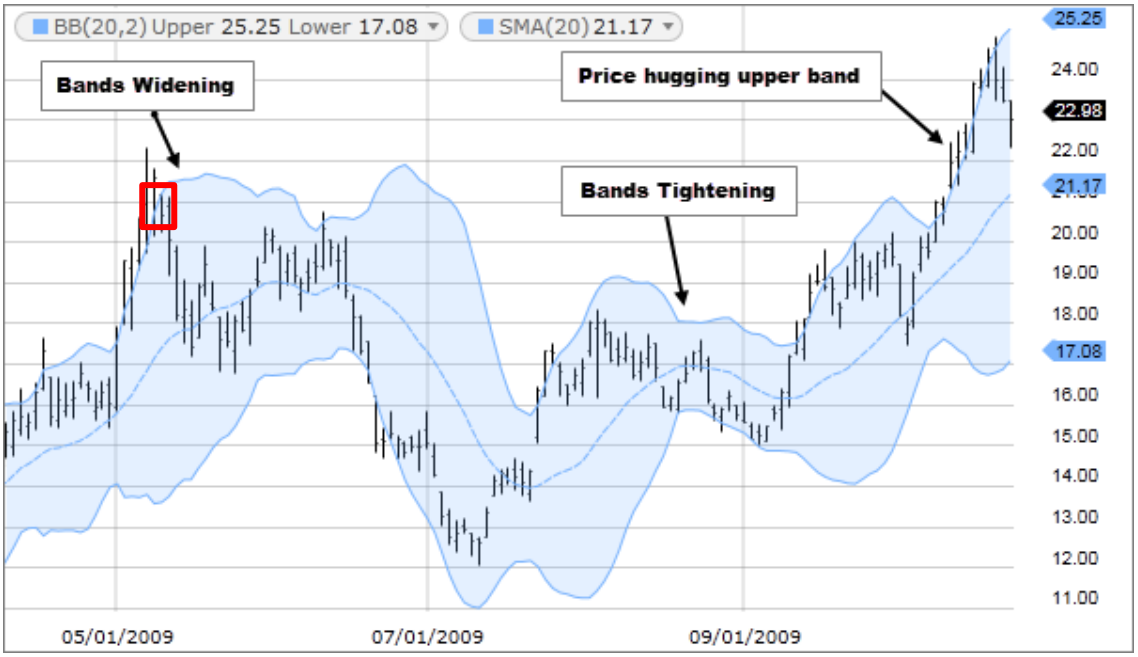
2) Technology Overview

Each of the theories in this section has significant variables and contributions relevant to the components in their respective formulas. In the BB, envelopes are plotted at a standard deviation from the simple moving average (SMA) or the current price. For long term stocks, the bands are at 2.5 standard deviations, for medium term it is 2, and for short term it is 1.5. Pertaining to the GFL, the formula is composed of components including performance, or the experience of a trader, skill, and breadth, or the resources to investing such as pecuniary access [7]. When referencing the components of the CAPM, it contains the market risk, risk-free return, and expected market return [12]. Finally, the APT, developed by Nobel-prize winning economist Stephen Ross has built a more complicated CAPM that introduced a framework explaining the expected theoretical rate of return of an asset, or portfolio, in equilibrium as a liner function of the risk of the asset, or portfolio, with respect to a set of factors capturing systematic risk [13]. Now, each of these variables is crucial to the concepts developed by each individual theory and formula. As a result of these variables and components, a correlation is drawn to how investors and traders approach the market. In the System Description section, a deeper description as to how these components are interconnected in each formula will be provided. These interconnections provide the composition of the formula and allows for conclusions and predictions to be computed not only by experienced investors, but potentially automated bots.

3) System Description

In this section, the components pertaining to each individual theory will express the correlation between the formulas and the conclusions drawn by each output along with what financial decisions are involved because of these end-results. As evident in figure 7.4, we see that each band is separated by a standard deviation band above and below of the smooth moving average line that falls between the two bands. This theory is critical because it indicates appropriate times to buy/sell stocks. For example, whenever the stock line touches the upper band from the outside going in, as shown by the red square in figure 11, it means the stock is at a higher price than usual and appears to possibly begin to drop, indicating it might be a good time to sell the stock.

Fig. 11 The Bollinger Bands [9]



In reference to the GFL, when figure 12 is analyzed, evidently the performance is measured by the information ratio, or the mean divided by the standard deviation. The skill is measured by the information coefficients that is in charge of correlating the forecasts to actual returns. Lastly, skill is multiplied by the square root of breadth. Breadth is recognized as the number of trading opportunities per year as a result of monetary resources.

$$IR \approx IC \sqrt{Breadth}$$

Fig. 12 Grinold's Fundamental Law [7]

CAPM is one of the most complex, yet most impacting model and theory to be introduced to the field of economics and finance. As evident in figure 13, the formula can be separated into two categories: 1) the risk-free return and the market dependent side that involves the market risk multiplied by the expected market return subtracting the risk-free return. This formula is essential because it implies that expected return is highly dependent on the market and is extremely difficult, if not impossible, to predict the future of stocks. However, it is a necessity to discuss this model because the APT built upon this formula and took more so the approach that the future of a stock can in fact be predicted when extending the quantity of variables that are being utilized. To further understand, E(R) is the expected return, R(f) is risk-free return, Beta is market risk, and E(R(m)) is expected market return.

$$E(R) = R_f + \beta * (E(R_m) - R_f)$$

Fig. 13 The Capital Asset Pricing Model [12]

Lastly, the successor of the CAPM, APT. APT uses the same principles as CAPM because it uses the idea of multiplying a risk factor Beta by the market return subtracting the risk-free return. However, as illustrated in figure 14, the APT improves upon the CAPM because instead of assuming there is only one market exposure variable that affects the expected return, APT says there is an unspecified number of macroeconomic factors that determines asset return [13]. For example, some beta factors are inflating, GDP growth, etc. In sum, the APT provides a more in-depth analysis to produce a more accurate expected return while still simultaneously agreeing that the biggest portion of the expected return comes from factors related to the market, which is outside of the individual traders' and investors control.

$$\begin{aligned} R_e = R_f &+ \beta_1 x (R_m - R_f) \\ &+ \beta_2 x (R_m - R_f) \\ &+ \beta_3 x (R_m - R_f) \\ &\vdots \\ &+ \beta_n x (R_m - R_f) \end{aligned}$$

Fig. 14 The Arbitrage Pricing Theory [12]

*C. **Interactive Brokers (IB)***

1) Summary

Interactive Brokers is an American multinational brokerage firm and operates the largest electronic trading platform in the country of the United States, according to the number of daily average revenue trades [16]. IB is relevant to the field of algorithmic trading because it also serves as a programming interface for executing orders. However, IB stands out from its peers because of its mature, flexible, and high-speed API and its significantly lower commissions in comparison to other trading firms [15]. Furthermore, IB is also one of the first brokerages to provide free access to algorithmic trading. The main principle of IB is that all human biases and psychological complications can be put away because of the change of responsibility being put onto the new and improved high-tech contemporary computers.

The IB programming interface requires either a Trader Workstation (TWS) or Interactive Brokers Gateway (IB Gateway) executing on your system to effectively run the IB program. In the system description section, a more in-depth analysis will be provided as to how the

interconnections for this system will work. For now, an understanding of a TWS must be grasped by overviewing the definition: TWS is the primary application for interacting with IB, and it provides an incredible wealth of features [20]. For the technological overview, the components and concepts of algorithmic trading will be outlined to provide a clearer picture as to what is needed to pursue using this programming interface.

2) Technology Overview

To begin, in order to utilize a programming interface, an understanding must be ensured as to what algorithmic trading refers to along with what components will be used to implement this algorithm. The idea of using a computer to place trades is called algorithmic trading [15]. The most important specifications and components for the implementation of IB can be categorized into three sections: 1) Turtle Trading System (TTS) by Richard Dennis, 2) Bollinger-MFI System (BB MFI) by John Bollinger, and 3) Support of Financial Information eXchange Computer-to-Computer Interface (FIX CTCI).

The Turtle Trading System is based on purchasing a stock or contract during a breakout and quickly selling on a retracement or price fall and is one of the most famous trend-following strategies [18]. In the most concisely written way possible, the TTS only considers making a new trade when one of its nine contracts price () rises above its 20-day high or fell below its 20-day low [15]. Similarly, the BB MFI follows a comparable concept in which was discussed to have two bands based on standard deviation above and below the average. These two formats for trading are implemented into IB using programming languages such as Python and C++. Furthermore, it also requires components and specifications such as an IB Server Interface, a TWS, and a client (the algorithm or application). Lastly, the IB also uses another interface framework: FIX CTCI. IB algo traders are familiar with the TWS API, but IB supports this lesser-known communication mechanism FIX CTCI that has the ability to support high-speed dedicated lines, extranet connections, and virtual private network communication [20]. In the System Overview section, a focus will be put on the TWS and how it is theoretically used when programming within the IB.

3) System Overview

In the execution of orders programmatically there three essential roles: 1) The algorithm, 2) TWS, and the 3) IB Server Interface. In figure 14 it is illustrated that all three levels to the execution of orders work together to allow for the IB to function as aimed. The interconnections are not only displaying arrows in one direction but implies that the implementation is of ambiguous direction. This allows for the application to tell the TWS when it wants to put in an order as well as allowing the IB servers to use TWS to accurately notify the application that a change has taken place. It is essential to pursue into further detail what exactly each of these roles contributes and is responsible for doing.

The algorithm is the program developed by the engineers that uses both forecasting and trading algorithms in conjunction. The forecasting algorithm uses data analytics of historical data alongside information feed to decide when the best time is to submit a buy or sell order of a stock. Meanwhile, the trading algorithm is the actual program dedicated to telling the TWS to purchase or sell the stock a given programmed time. The TWS is essential because it is the liaison between the program and the client's portfolio. Without the TWS we simply have lines of code and a stock portfolio waiting for action. Lastly, the application, or client, is the algorithms that have been composed based off economic principles and predictions from theories presented in the past; thus, avoiding psychological inclination as a result of natural human bias by placing the responsibility on the program to make the decision off of statistical and mathematical computations.

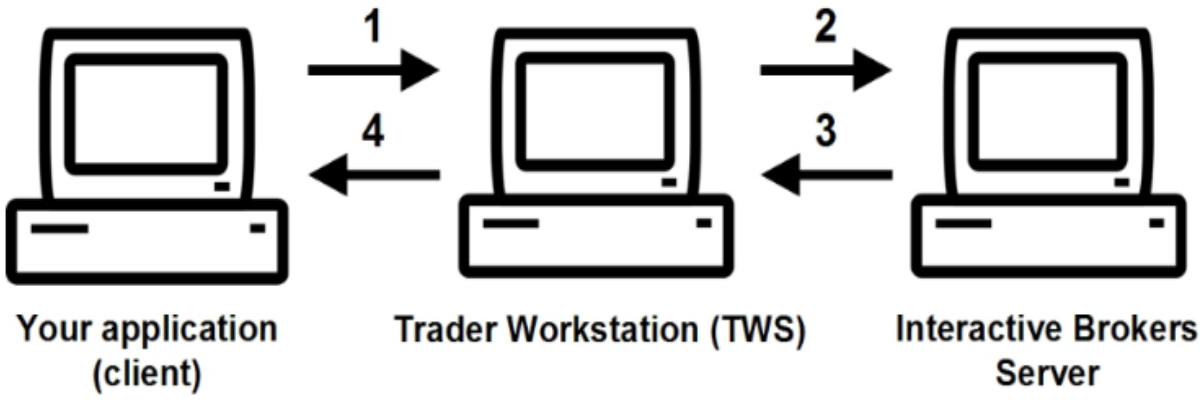


Fig. 15 Executing Orders Programmatically [15]

In conclusion, these three sections clearly outlined the composition, significance, and impact of using statistical/mathematical computations, remunerative/financial theories, and interactive brokers into the field of machine learning and economics. More importantly, their approach provided us with certain guidelines and a variety of approaches to implement into the design phase for the Stock Market Trader Bot. The system plans to combine key components discussed in this background section of excellent theories and projects while further building upon them and applying its crucial concepts to the fields of artificial intelligence and financing. The computing behind Hedge Funds allows for the appropriate framework and idea introduction as to how the Bot can apply machine learning to simulate the actions of a trader while also utilizing historical pricing data along, forecasting, and trading algorithms to optimize a portfolio and pursue an automated profitable experience to users. Furthermore, the Bollinger Bands and Grinold's Fundamental Law gives participating engineers of the Stock Market Trader Bot a complex and detailed visual as to how stock's data analysis can use mathematical derivations such as standard deviation and breadth to train an automated algorithm to understand and predict what are the best, most appropriate time to buy and sell orders. Lastly, the interactive brokers are huge in its contributions to providing the stake holding engineers with the correlating dos and don'ts with respect to algorithmic and development phases, ensuring that this model applies the best of all worlds while simultaneously promoting originality and most importantly a uniquely, profitable trading experience for users of all varying extensions of economic and financial knowledge.

IX. Intellectual Property Considerations

Avoiding intellectual property infringements is aa requirement for the project. By researching patents and copyrighted works that may present legal conflicts in the project, efforts can be made to change the product to avoid infringement beforehand rather than later. In addition, respecting others' work contributes to maintaining a safe and collaborative environment for designers, including ourselves. This section presents patents and copyrighted works that may prevent us from using certain functions in the stock trader bot.

A. System and Method for a Risk Check

The inventor of this patent is Sagy Pundak Mintz from Austin, Texas. The patent was issued on January 4, 2022, and is described below [22]:

1) Summary

In electronic trading, the patent encompasses a series of operations that a processing unit executes. The processing unit first obtains a trading strategy containing quantities of tradeable objects (in the case, stocks). The processing unit has risk values or derives a risk value from a pre-configured maximum/minimum value set by the trader. The processing unit then decides if the trading strategy is acceptable by using the risk values and the trading strategy sent to the processing unit. Finally, the processing unit sends an electronic order to a market based on its decision.

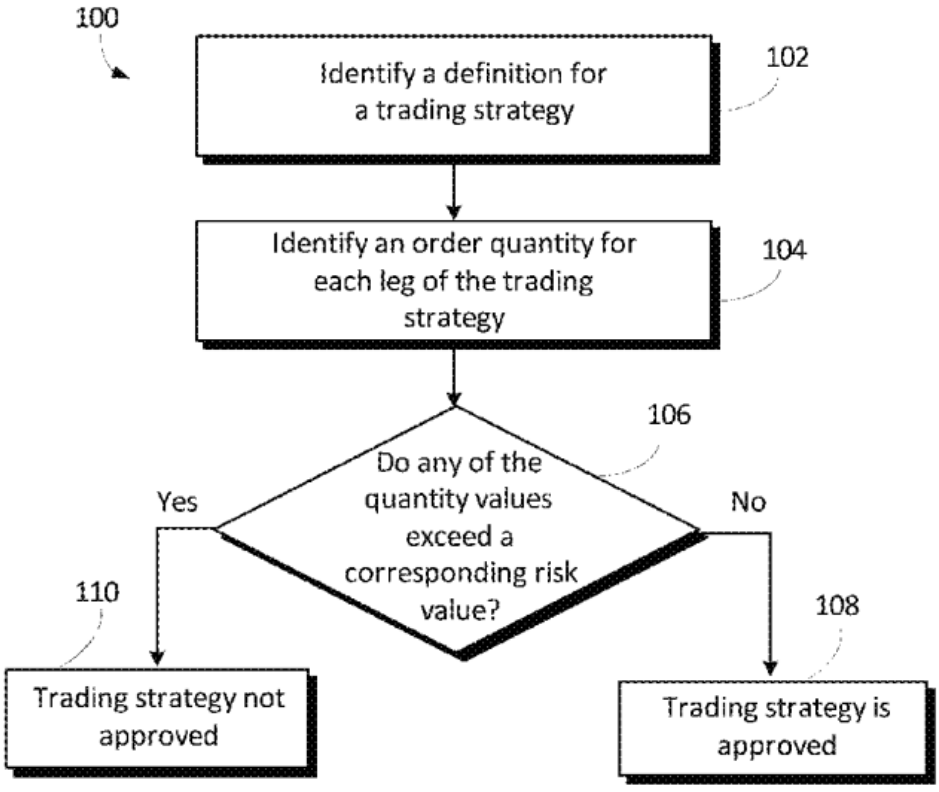


Fig. 16 Diagram of Patented Function [22]

In addition to having the computing device decide, the patent also encompasses a server-side decision. In the server-side case, a server receives a trading strategy, calculates risk factors, or has pre-set maximum/minimum values, and then still approves/disapproves of the trading strategy implemented; an example is shown below.

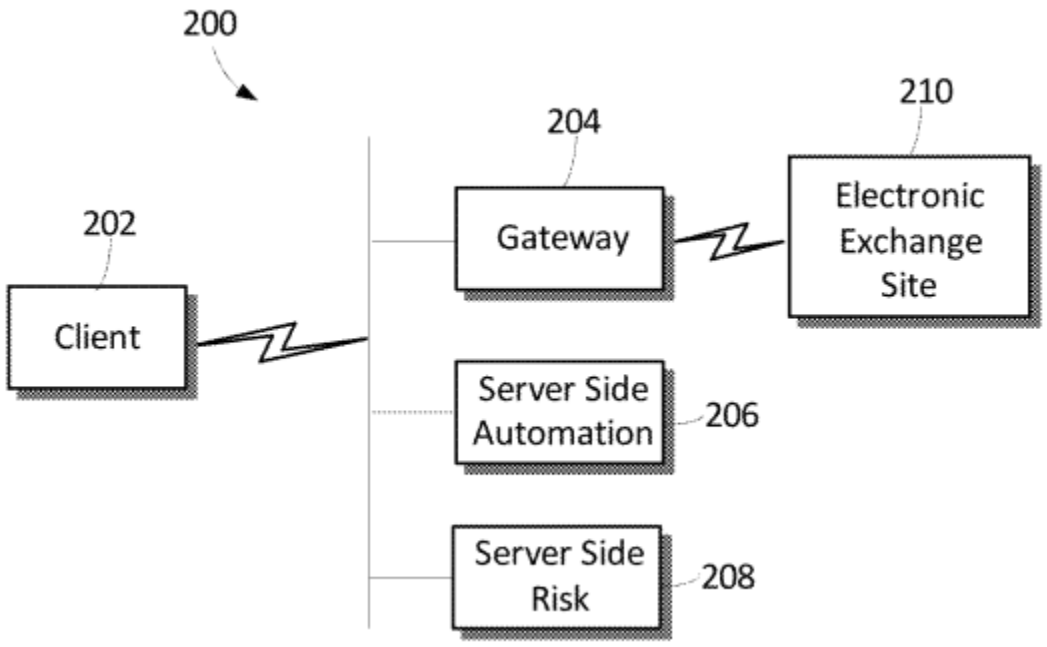


Fig. 17 Diagram of An Example of Patent A [22]

2) Claims Summary

This patent contains 16 total claims, with 14 claims being extensions of the other two claims:

- Trading strategy received by a processing unit that:
  - Includes one or more quantities for tradeable objects and
  - Determines if the trading strategy is acceptable based on an associated position risk value or
  - Determines if the trading strategy is acceptable based on the associated position risk value pre-configured **from a trader** and
  - Executes the trade based on the decision-making in the last two bullet points

3) Non-Infringingment Explanation

While this patent may seem to cover what the project aims to be, the project does not violate any of the claims. This patent covers only decision-based trading from risk values associated with the tradeable objects (stocks). Because the algorithm aims to predict the future of stores, we will not be taking said risk values into account. In addition, the algorithm will not use the stock's pre-configured maximum or minimum values from the trader. In short, because the algorithm avoids utilizing the user's risk factors and maximum/minimum values, the algorithm we will develop does not infringe on the patent mentioned.

B. Methods, Systems, and Computer Program Products for Trading Financial Instruments on an Exchange

This patent was invented by Gary L. Gastineau, Todd J. Broms, Daniel J. McCabe, and Paul E. Kuhnle. It was granted on February 24, 2009.

1) Summary

The patent covers a system, methods, and computer program products for trading financial instruments listed on an exchange. The technique is described as follows: an order for buying/selling a financial instrument is received, along with at least one alternative option for the order. The alternative can be price-based or volume-based. Next, another charge matches the wants of the first order and its alternate options, which are also calculated, and then both orders are executed.

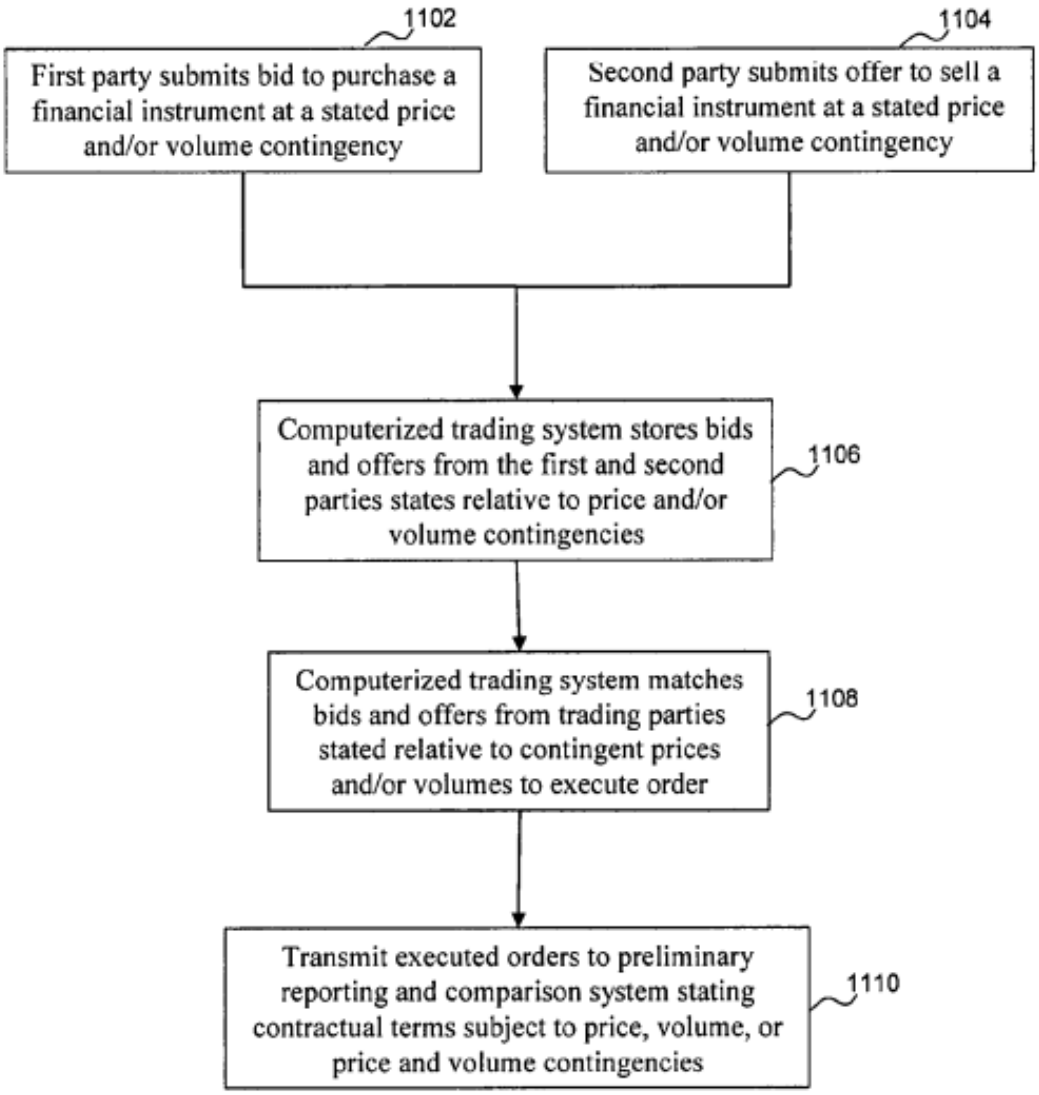


Fig. 18 Summary of Process of Trading Financial Instruments [23]

The exchanger then reports the trade and results to at least one relevant party, meaning one of the users, a broker, the market reporting system, etc. Through this method, the invention creates a safe and secure environment for the traders and gives the exchange more of a chance to happen, as there are alternate options that the traders consider.

2) Claims Summary

The patent holds a total of 79 claims, many of which pertain to different implementations of the patent; however, the base operations remain the same:

- A computing-based method for trading exchange-listed financial instruments that:
  - Executes the trade of said financial instruments
  - Receives an order to buy/sell a specified financial instrument relative to its net asset value, along with one or more alternative order options
  - Receives another order to sell/buy the specified financial instrument, with the same options allowed to the first order
  - Executes the order and calculates a middle ground for the two orders
  - Reports the order and delivers the information to at least one interested party, including the two clients and exchange participants or the exchange itself.

3) Non-Infringement Explanation

This patent is related to the project because it participates in an exchange and handles an operation for at least one exchange user. However, this patent encompasses exchanging two users' financial instrument properties and doing so safely and securely. The project will be dealing specifically with the stock market intending to bring a profit to the user, and instead of trading between two peoples will be trading on the stock market itself. Therefore, the project does not violate this patent's intellectual property.

C. System and Method for Order Placement in an Electronic Trading Environment

The inventors of this patent are Michael J. Burns, Eric M. Herz, Sagy Pundak Mintz, Alexander D. Dietz. It was granted on April 29th, 2014.

1) Summary

This patent holds a system and methods for facilitating order management on the market. The method and system are described as taking orders at a specific calculated price while taking in other orders above and below the calculated price. These orders are separated into "Legs," and the system keeps track of which are below and above the calculated price. This way, if the market were to fluctuate, there would be orders readily available to take the place of any order at the original price. If the market price for trade fell, a leg order found at the new price could be used to trade.

The figure below illustrates the process where N orders are placed in the system, and the desired price is set for the orders spread within the system. Then the system calculates the costs of each leg based on market conditions and places the N orders in the legs according to the prices set in each leg. As seen in the decision-making section, if the order is a buy order, it is placed into a lower value leg; if it is a sell order, it is placed in a higher-valued leg. This way, it also maximizes the equities of the buying/selling parties.

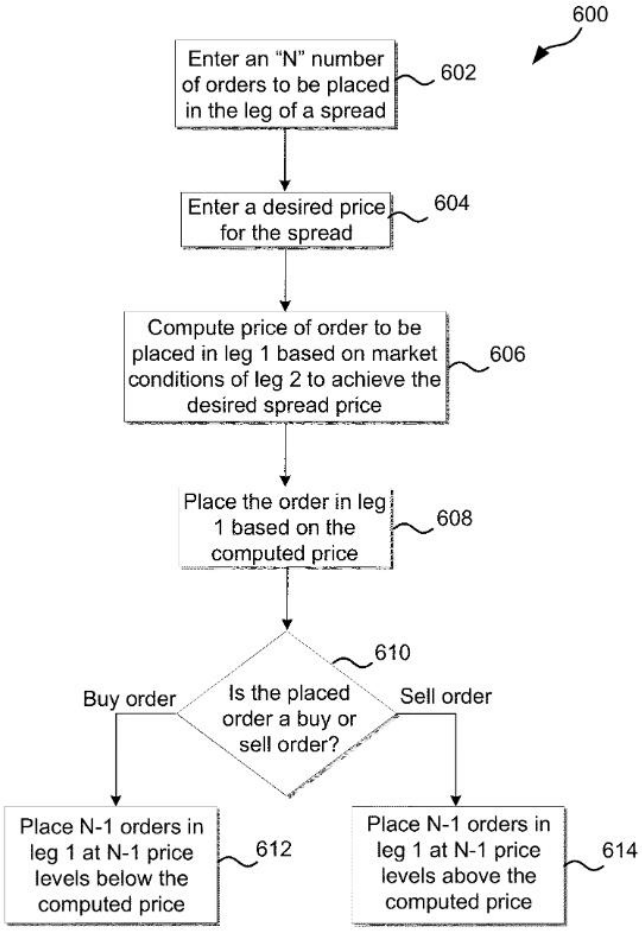


Fig. 19 Basic Overview of Electronic Trading Environment System [24]

2) Claims Summary

This patent contains 11 claims, with all but one referencing another claim.

- A computing device that
  - Calculates a price for the first order based on market conditions and stores the order and price
  - Creates a range of prices and sends orders within that range
  - If none are found, recalculate prices and repeat the process

3) Non-Infringement Explanation

While this patented invention involves a trade market, stores data within the market manages orders, and recalculates prices based on changes within the market; the project does not violate any of the patent's claims. The project does calculations involving the market; however, it does not facilitate the transactions within the market but instead handles the transactions for a user or advises a user to create a profit. Therefore, we are found not to violate the patent's claims as the algorithm will not be sending orders within a range of prices and then recalculating the market price.

To conclude, this Intellectual Property and patents section is important because it allows us to have a clear understanding of the patents that are relevant to the product. Furthermore, by going into detail and researching such patents, the team was able to effectively prepare to avoid infringing patents and secure a safe, non-infringing plan for the product.

X. GLOBALIZATION

For all high impacting and truly successful engineering projects, a heightened importance has been emphasized on its globalization outcome specific to its triumph in international and intercontinental grounds. When discussing assumptions for our Bot, it was stated that this product would be focused on the U.S. market. However, our product is based on sophisticated trading algorithms that use historical data analysis and proven economic theories to formulate a psychologically uninclined Bot to make profit ensuring trading decisions on all types of stocks, options, and more, regardless of the current market location. Therefore, because our product will have a strong backbone of decades worth of economic science and data science driving its profitable algorithms, it will be easily integrated worldwide in any and all stock market exchanges whether it be the New York Stock Exchange located in the Americas, or foreign markets such as the London Stock Exchange, Hong Kong Stock Exchange, etc. The Stock Market Trader Bot is destined for success in its contemporary market as a result of its easy working user interface, predictive, and profitable decision-making algorithms. However, it is only a matter of time until globalization of this profit-bound product is widespread to the extent that it is used internationally across all market exchanges.

Participating engineers of this intricate Bot product must maintain an open mind to demonstrate global awareness about intercultural and global issues that may influence the implementation of our design. Because our project centers around programming and software development, the team members educated themselves on the effect technology has on intercontinental structures and cultures in order to maintain global recognition of issues centered around the market our product will be relevant to. The rise of technology and software is inevitable and has proven to only be an endless competition of self-improvement. According to CompTIA, the tech industry composes of over 35% of the total worldwide market [30]. From this vast percentage of tech involvement, by the year 2030, Artificial Intelligence applications and software such as those of the Stock Market Trader Bot, is expected to contribute over 15 trillion dollars to the economy around the globe [31]! Technology has provided a direct path of expansion and positively driven change in the aims to improve the world for a better tomorrow. Nevertheless, as globally aware engineers, the team understands that these worldwide technological advances and accomplishments would have been impossible to occur without global involvement and cooperation. This comprehension that global integration and collaboration directly affects a successful project design allows for a victorious, result driven Stock Market Trader Bot that can be utilized across the world regardless of cultures and international circumstances that may arise.

To promote appropriate globalization, each individual engineer conducted a Global Perspective Analysis to apprehend the need of this product in international and intercultural markets. Furthermore, this would encourage global engagement by analyzing international standards and need fulfillment that the product would be covering for diverse user groups of different cultures nationally. The Stock Market Trader Bot nurtures and advances global engagement because it seeks pecuniary profitability in all markets that is centered on a specific currency. This Bot will contain algorithms that are compatible with different types of trades that use international currencies, ensuring there is active globalization compatibility through the entire developmental process for the product. In the rest of this globalization section, the importance of the World Trade Organization (WTO) in international markets, trading barriers in global markets, collaboration tools for traveling team members, and this product international success.

A. World Trade Organization (WTO)

This subsection of the WTO aims to focus on depicting a correlation of the WTO with general engineering standards. The WTO operates the global system of trade rules and helps developing countries built their trade capacity. It also provides a forum for its members to negotiate trade agreements and resolve trade problems faced by conflicting countries [32]. Because the Stock Market Trader Bot is aimed to work with trading and varying currencies, the WTO will be directly correlated to the developing nature of our product. The WTO is known for improving the lives of developing countries, negotiating trade rules, overseeing global agreements, maintaining pen trade, and settling disputes [33]. The WTO will ensure legality of the globalization of any project/product potentially including this Bot. Moreover, they are the only global international organization dealing with the rules of trade between nations. At the core is the WTO agreements, negotiated and signed by the bulk of the world's trading nations parliaments [34]. Overall, the main objective is to assist creators of goods/services and exporters/importers in business conduction, directly correlating to the production and business marketability nature of the Stock Market Trader Bot.

B. Trading Barriers

When indulging in the components of trading, specifically global trading, there are factors and obstacles, known as trading barriers, preventing products from being used worldwide as originally intended. This section highlights the significance of eliminating the barriers to trade and how the selection of standards will allow the Bot to compete in a Global Market. To ensure wide spreading the Trader Bot's globalization, the implementation of the product must be maneuvered in a manner that avoids confrontation with trading barriers respective to the fields of artificial intelligence. Circumvention of artificial intelligence and machine learning trading barriers will be accomplished by complying to both local and global standards outlined by the Institute of Electrical and Electronics Engineers Standards Association (IEEE SA) in collaboration with the WTO.

In efforts to minimize trading barriers for our automated Stock Market Trader Bot, this team aims to follow the IEEE Std 1872.2-2021 specific to Automated Bot Ontology standards. This set of active standards is focused on following international standards compatible with the WTO for Ontologies for Automation and Artificial Intelligence is an extension of IEEE Std 1872-2015 [35]. This standard represents additional domain-specific concepts, definitions, and axioms commonly used in automation and how projects in this field should strictly follow such standards to promote facility in trading globally [35]. For example, to specify the domain-specific concepts needed to unambiguously describe the design patterns of the Bot, we identified that this product is specific to global countries that have a Stock Market Exchange and allows for automated trading. Finally, the IEEE Std 1872.2-2021 pushes for specific autonomous systems consisting of a Bot to operable in various environments [35]. The Stock Market Trader Bot follows this standard by being operable cross platform in varying operating systems to minimize trading barriers.

In conclusion, it is important to have a set of guidelines and standards to follow for a properly globalized product that bypasses trading barriers. This system ensures to follow specific engineering standards outlined by the IEEE SA and WTO to provide proper international globalization in a manner that nurtures usage in varying cultures, trading and operating environments.

C. Collaboration Tools

The Stock Market Trader Bot group has four participating engineers. For the duration of the design and implementation of the product, collaboration tools to stimulate communication amongst

members has been an emphasis. The team members actively partake in a WhatsApp group chat where weekly reports, updates, and task management is a priority. When sharing significantly large files amongst members, Google Drive has also been used to provide a shareable link for viewing. The most significant collaboration tool for the product would be a version control. Because the Bot is almost completely reliant on software, to allow for easily manipulatable programming the team utilized Git and GitHub to actively analyze changes in coding as well as maintaining updates in a shareable manner amongst participating engineers. All tools mentioned (WhatsApp, Google Drive, Git, GitHub) are cloud-based tools which ensures that demographics and location will not be an issue when developing the product. By utilizing these cloud-based tools, the team members are guaranteed to have a smooth collaboration even when a member is traveling.

D. International Success

An imperative question with respect to the Stock Market Trader Bot's development is whether or not the product is destined for international success for markets around the globe. Not all countries in the world have a Stock Market Exchange, however, for those that do, the team decided to seek a final product that would promote appropriate compatibility success. With respect to these countries worldwide that have their own stock exchanges, trading algorithms would surely only merely provide beneficial outcomes. Wherever a stock exchange market exists, the demand for a profitable, algorithmic trading bot is guaranteed to be high. To test this exact statement, participating engineers contacted students and faculty from universities in other countries and introduced the concept and proposed final product of the Stock Market Trader Bot. The aimed outcome of seeking international approval from these contacts is to understand and learn more about their perspectives of the Bot in their respective countries, how they feel it will be accepted in their country, and whether or not it will be successful.

Elianys Nicado Leal is a current Ph.D. seeking student from Mexico majoring in Chemistry who has proven to have an interest and background in trading within the Mexican Stock Exchange (Bolsa Mexicana de Valores). Proceeding the introduction of the Stock Market Trader Bot to Elianys, she immediately spoke on the rise of technology she has experienced in her university atmosphere and how her fiancé, a current programmer, is consistently emphasizing the growth and potential of machine learning algorithm applications. Having strong interests in the economy and trading, Elianys expressed that the Stock Market Trader Bot would definitely attract traders in her country and be easily successful if it was backed by economic theories and an emotionless Bot. When prompted of the discussion of the Bot's effect on culture, she said that it would not be an issue for Mexican culture and that it would surely be a product that would become widespread amongst novice traders and civilians in her country.

Marla Milanes Molina is a recent graduate of medical school from Cuba who also spent time studying abroad in South America. The objectives and overall concept of the Stock Market Trader Bot was introduced to her and was followed by questioning her thoughts and perspective with respect to the product. Following the introduction, Marla began to express her firsthand experiences of the recent growth of automation she saw in the different university curriculum. Although Cuba does not have a stock exchange market, in her time studying abroad, Marla said it was evident that she felt it was overwhelming to enter the Stock Exchanges as a novice trader with trading interests but little experience. She expressed that the Trader Bot seemed like an extremely easy and smooth way for her to enter the stock market as a trader. Furthermore, Marla stated that there would likely be little to no conflict with respect to intercultural issues. She felt that in both South America and Cuba, there would definitely be a demand for the machine learning algorithms along with the

trading Bot. Marla specifically stated that the benefits of using the product would be heightened as a result of the easy user interface and ability to provide an opportunity for a profitable experience.

In conclusion, candidates from countries located in the Caribbeans and South America expressed that the future lies in automation and machine learning. Therefore, our Bot would follow such technological trends in a manner that would attract financial gain, a goal of all traders. With respect to cultural conflicts, these contacts ensured that this aspect of globalization would be of no issue. Finally, these candidates mutually agreed that intercontinental and global success would be nearly guaranteed if this product was to be pursued worldwide.

Overall, Globalization is evidently a crucial component for the team and final product because of the demanding procedures necessary to take for appropriate project development. To ensure a marketable, intercultural, and internationally compatible product, participating engineers must consider the imperative actions needed to promote globalization within the Stock Market Trader Bot along with potential issues that may arise globally, thus promoting global awareness amongst team members. This section analyzed the effects of the WTO on our product, identified trading barriers that would provide active competition in the Global Market, Collaboration tools by team members, and specific examples of real-time university students that backed the idea of international product success.

XI. STANDARD CONSIDERATIONS

Standards are an essential component of every project since they ensure the quality and reliability of the project and its goals. Always keeping these standards in mind is important to the project to confirm the correct design of the product. Standards also make the product more acceptable to communities and companies alike. Without considering these standards, the product may not be considered for creation or would be rejected by manufacturing due to lack of compliance and quality. Furthermore, the trust and design of the product may be hindered due to not having any standards to adhere to. Luckily, there are credible organizations and companies that implement and create these standards for engineers, designers, and manufacturers. Without these organizations, there would be a lack of trust and quality in products being designed throughout the world as these standards uphold the integrity of these projects.

A few organizations that create and uphold these standards consist of the International Organization for Standardization (ISO), the International Electrotechnical Commission (IEC), the American National Standards Institute (ANSI), the International Telecommunication Union (ITU) and the Institute of Electrical and Electronics Engineers (IEEE). These organizations are trusted around the world and set the standard for not only safety, but quality and assurance of products with their labels on them. Setting different standards for products to make them efficient and effective is part of their job, as well as reading through current set standards for electrical devices such as sensors, batteries, power consumption devices and more. The standards for these devices are reviewed and then revised to adhere to global consumer needs and industries. Non replication of standards is important for these organizations, as is constantly revising and improving all global standards for devices as well.

For the Stock Trader Bot to be successful, the team will adhere to a set of standards and will properly select the standards necessary for the project to be realized. With the assistance of a vast number of organizations that have standards already in place, the team is able to choose which standards are best fit for the team's project. The team will meet with a set of these standards to uphold the reliability, quality, safety, and integrity of the product. For this, the team must acknowledge and inspect the types of technologies it will be using, followed by the compliance of the standards set below:

A. 12207-2017 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Software life cycle processes

The first standard establishes a common framework for the software life cycle process. Activities, tasks, and processes of software systems are applied directly within this standard to produce, supply, develop, operate, maintain, and dispose of software products. This is accomplished through the willingness of stakeholders with the ultimate goal of producing sufficient customer satisfaction. This standard applies to internal or external segments of an organization and includes the portion of firmware involved. Defining, controlling, and improving the software life cycle process is also included within this document as it pertains to all organizations and projects alike.

The main purpose of this statement is to "define a set of processes that aim to facilitate communication among acquirers, suppliers, and other stakeholders in the lifecycle of a software system" [40]. This standard also applies to many fields including the conception, development, production, utilization, support, retirement and to their acquisition and supply of the software system. "The life cycle process can be applied concurrently, iteratively, and recursively to a

software system and incrementally to its elements" [40]. Types of systems this standard can be applied to include one-of-a-kind software systems, software systems for commercial use or public distribution, customized and adaptable software systems. This includes complete stand-alone software systems and systems that are embedded and integrated into a larger and more tailored to system. Relationships between software elements are illustrated below:

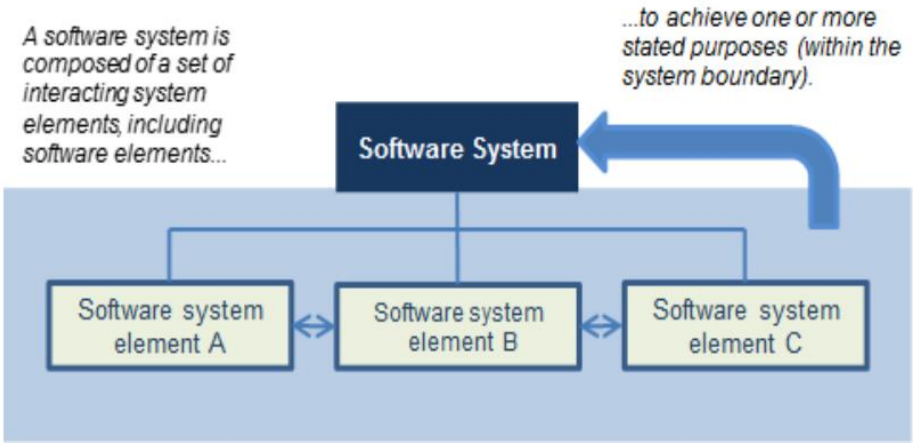


Fig. 20 Software System and Element Relationship [40]

Relationships between software elements are imperative to a system's design and can typically be represented as shown above. Approaches to software activities include decomposition, object-oriented approach, where the system elements are laid out in a non-hierarchical diagram, and a prospective system element approach. In the matter of a prospective system element approach, the system life cycle processes are applied recursively to a system-of-interest and resolves its structure to the point where understandable and manageable software system elements can then be implemented.

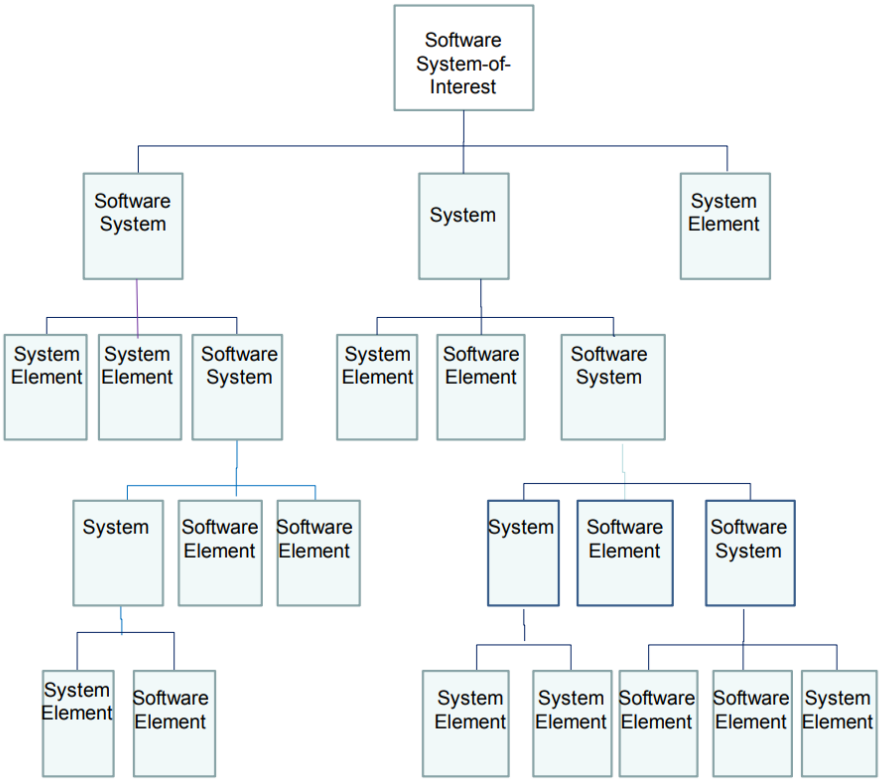


Fig. 21. Example of Software System-of-Interest Structure [40]

Figures 20 and 21 imply a hierarchal relationship even though realistically there will be many systems that do not contain a hierarchy such as networks or distributed systems.

B. ISO/IEC/IEEE 14764 - International Standard - Software engineering - Software life cycle processes – Maintenance

This standard is one that describes an iterative process for managing and executing software maintenance activities. The standard itself is not restricted by size, complexity, criticality, or application of the product and uses a process model to depict aspects of software maintenance. Planning and executing software maintenance while under development is the main subject of criteria for this standard. In an ideal situation, maintenance planning should begin during planning for the development itself. This standard provides a framework that generic and specific software maintenance plans can be evaluated to tailor to the maintenance scope and magnitude of software products. A framework is provided, as is precise terminology, and processes to allow consistent application of technologies for software maintenance. The basis for the maintenance process and its activities are consistent with ISE/IEC/IEE 12207:2017, Systems and Software engineering via software life cycle processes. The standard defines the activities and tasks of software maintenance and provides maintenance planning requirements. What the standard does not address is the operation of software and operational functions such as backup, recovery, or system administration which is assumed to be performed by those who are creating the software itself.

Maintenance is a technical life cycle process that is performed during the life cycle of software according to (ISO/IEC/IEEE 12207). “When acquiring and supplying the lifecycle process, it may initiate the process implementation activity of the maintenance process through a modification request or problem report” [41]. The operation process calls for maintenance through a

modification request or report of an issue. The maintenance life cycle is then invoked, and the technical life cycle is implemented to develop any required changes as part of the maintenance strategy. The figure below shows the relationships between the maintenance process and other maintenance related processes including the elaboration of the maintenance process into its own activities.

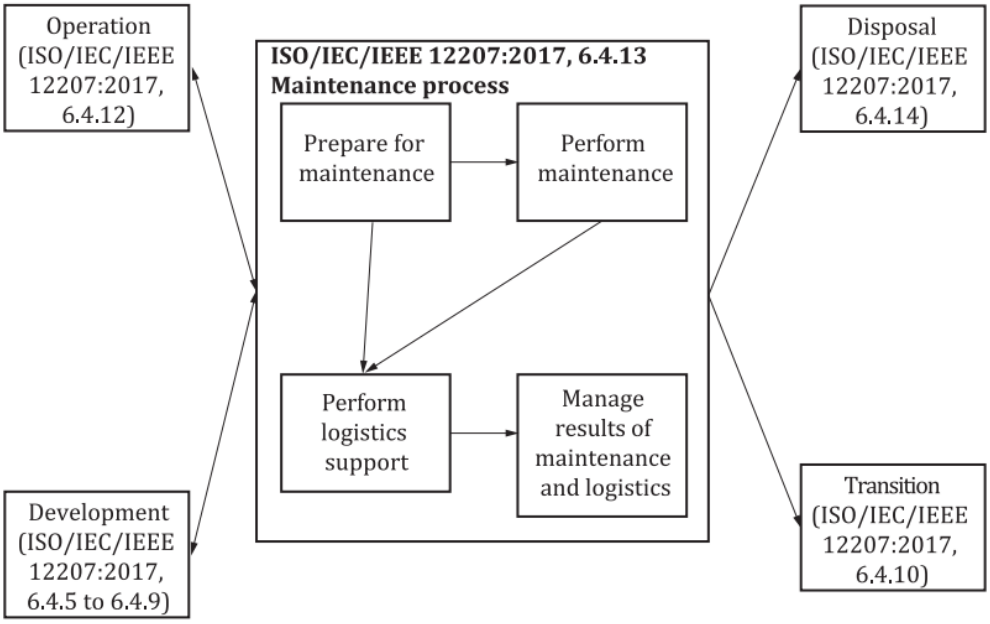


Fig. 22. Relationship Between Maintenance Process and Other Processes of ISO/IEC/IEE 12207 by Clause Number [41]

As a result of the successful implementation of the Maintenance process:

a) Maintenance constraints that influence system requirements, architecture, or design are identified.

b) Any enabling systems or services needed for maintenance are available.

c) Replacement, repaired, or revised system elements are made available.

d) The need for changes to address corrective, perfective, or adaptive maintenance is reported.

e) Failure and lifetime data, including associated costs, is determined.

C. ISO/ IEC/IEEE 29119-1 Software and systems engineering — Software testing

The purpose of standard ISO/IEC/IEEE 29119 is to provide a set of standards pertaining to software testing to be utilized by any organization when performing software testing during any life cycle. Since there are so many types of software, software organizations, and methodologies the ISO/IEC/IEEE 29119 series was created to adhere to all of the following software development strategies: Object-oriented, traditional, agile, and DevOps. “Software domains that are included within this standard consists of Information Technology, Personal Computers, embedded, mobile, and scientific classifications” [42]. A software testing quality management description is provided for the verification, validation, and implementation of static and dynamic forms of testing. Testing plans and strategies are provided with a context of risk-based testing which is the recommended approach to managing testing. Levels of testing, test types and design techniques will be described with the context of their inclusion as part of the generalized testing strategy.

As a brief introduction to software testing, this standard provides a description for the relationship between quality management and verification and validation. Quality management involves quality assurance and quality control with the notion that quality assurance is focuses on proper implementation while quality control focuses on activities supporting the goal of meeting appropriate levels of quality. In terms of verification and validation, these are two separate entities that employ testing as their principal practices. Verification is conformance testing with specifications focused while validation is focused on the acceptability of a test item to meet the needs of stakeholders. Static and dynamic testing supports verification and validation as shown below:

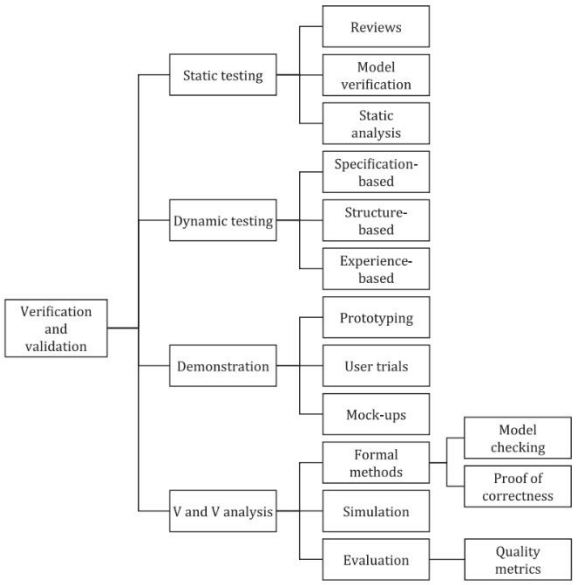


Fig. 23. Verification and Validation

After thorough review of these standards and summarizing their main points, the team will adhere and rely on the standards chosen when considering the design process of the project. These standards are imperative for the technologies the team will be using especially since the main focus of the project will be software oriented. Software based standards provide plentiful information about the design process and standards set across the globe especially by the IEEE, ISO, and ISE organizations. These organizations aid in providing a sound structure for groups and individuals alike to produce proper and well-designed projects. In conclusion, the team will adhere to the following standards chosen with regard to the project:

- 12207-2017 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Software life cycle processes
- ISO/IEC/IEEE 14764 - International Standard - Software engineering - Software life cycle processes – Maintenance
- ISO/ IEC/IEEE 29119-1 Software and systems engineering — Software testing

XII. HEALTH AND SAFETY CONSIDERATIONS

Health and safety are an imperative part of the overall design process. This section will take into consideration the health and safety of participants of the project and users of the product. The World Health Organization defines health as “a state of complete physical, mental and social well-being and not merely the absence of disease or infirmity”. Safety is important because maintaining an environment that is not only hazard free but mentally safe to be in is essential to the designers and users alike. Companies apply an SMS (safety management system) to ensure the safety of the workplace and its workers. This system works in a way such that precautionary hazards are brought to light to mitigate any future hazards that may occur. The team will take these hazards into consideration and apply precautionary hazard measures when designing the product. If a product puts the safety of a person at risk, this is something that should be dealt with immediately and is a top priority when considering health and safety. A person’s health and safety are an imperative part of a products design and could affect the outcome and manufacturability of a product in general.

Our project, the stock trader bot will be designed with health in mind. The team will especially take mental and social well-being into consideration when designing the bot. The bot will be dealing with people’s money, therefore their earnings and hard work. This will be taken into consideration when dealing with losses as well as gains by the bot’s performance. Design of the bot the bot is in itself, a daunting task that should be split amongst the group members for the benefit of each team member’s mental health. When implementing the design of the stock trader bot, these factors will be taken into consideration and the team will ensure a safe and healthy product that is being created.

*A. **Team Safety***

Guidelines will be set to ensure the safety of the team members and to provide a stable yet comfortable place of work. Since the team will be working remotely, a quiet and non-hazardous environment should be set in place for each member to make their contributions. If the health and safety of any team member were to be compromised, immediate action will be taken in the form of assisting and assessing said team member. If any hazards or health risks should begin to ensue, it is the responsibility of each team member to communicate clearly to one another of any possible risk. The standards regarding the safety and health of the team should be strictly adhered to and will be listed below:

1) Team members should be working in an environment that is non stressful, comfortable, and safe with enough breaks throughout the day. This ensures a productive, stress-free area of work that maximizes team safety as well as efficiency.

2) Team members need to be aware of their own surroundings at all times. This includes wearing proper attire and protective gear which ensures the safety of each team member.

3) Team members must be aware of their stress levels and mental health. This is imperative for the team’s success in the long run as if one or more members of the team become overwhelmed, the entire project could be in jeopardy.

*B. **Labor Safety***

The overall goal of the stock trader bot has been to make profit for its users. Not only this, but ensuring that team members working on the bot have a safe environment to be in. For this reason, having an open line of communication between the team members is imperative and achieved through mobile communications. To minimize risk of bugs or non-compliance of previous standards in the bot, the team will consistently meet up and discuss the code that has been written for the week. The team will also use version control systems such as Git and GitHub to be able to work on the project through independent branches. This means that if any compliance were to be broken or bugs were to be introduced, the team can always revert back to a previous version of the code that adheres to all considerations.

Preventing burnout can be a vital aspect to a project especially in software development. There is a high probability of the team having to work for many hours to achieve the goal that has been set by its own standards. This can be minimized by keeping a log of worktimes that each member has committed to the project ensuring that equal amounts of work is being spread out evenly.

In all, the health and safety considerations of a project is vital to its success. Preventing hazardous work environments, burnout, keeping stress levels low, having a strong team mentality, and ensuring a safe, productive environment are all essential to a project's outcome. This topic ensures that both the team and the users of the product will be safe and healthy when both designing and engaging with the product. Any aspects of the project that can be deemed as unsafe or hazardous can create faults within the product and may eventually lead to a general failure of compliance. If this were to happen, the code of ethics could be breached, and the manufacturability of the project could become nonexistent. The stock trader bot project will adhere to all above set standards and always take the health and safety of its team members and users as top priority.

XIII. ENVIRONMENTAL CONSIDERATIONS

It is crucial to analyze any lasting impact a project may have on the environment. This section briefly outlines the process to be taken in order to promote environmental considerations and ensure the Stock Trader Bot does not negatively impact the environment during the design and implementation phases. Furthermore, it is undeniable that in recent years there is of heightened importance with respect to how different engineering inventions affect the health and longevity of environmental resources. It is because of this importance that more corporations and large companies avoid any potential harm to the atmosphere and world around us to effectively consider the environment. It is essential as engineers to increase the levels of environmental awareness and all-around care for the effects a project may have. As a result of this generational movement toward a more environmentally aware group of engineers, the team has decided to delve into the environmental considerations with respect to the Stock Market Trader Bot.

The Stock Market Trader Bot is a Software dominant product that does not cease to directly impact the environment. Moreover, the Bot is relevant to fields of trading and financing, meaning its potential affects of profitability or financial loss is of no direct concern or impact to the environment. Nevertheless, in the rest of this section, the environmental considerations will be broken down into subsections of the Restriction of Hazardous Substances Directive (RoHS) and how our product is relevant to it, a disassembly subsection to discuss how easily our product can be disassembled, a Hannover Principles subsection and how it pertains to this Bot, and lastly, the Life Cycle Impact Assessment for our project.

A. Restriction of Hazardous Substances Directive (RoHS)

The RoHS is relevant to the entire electronics industry because it serves to ban ten hazardous materials that was often used in previous electronic models and is evident to negatively affect the environment. Furthermore, as a direct consequence of this directive, products cannot be released into market unless they comply with the RoHS directive. However, because our product does not require any hardware, nor does it require electronic materials, it does not conflict with this directive. Nevertheless, although our product is completely Software based, as growing engineers the team members must still be aware of these ten hazardous materials in the rare case it is to be incorporated into an electronic project:

- Lead (Pb)
- Mercury
- Hexavalent Chromium (Cr VI)
- Polybrominated Biphenyls (PBB)
- Cadmium (Cd)
- Polybrominated Diphenyl Ethers (PBDE)
- Bis (2-Ethylhexyl) Phthalate (DEHP)
- Benzyl Butyl Phthalate (BBP)
- Dibutyl Phthalate (DBP)
- Diisobutyl Phthalate (DIBP)

In conclusion, the team has decided to refrain from using these materials in its entirety with respect to the designing of the contemporary and future models of the Trader Bot. Regardless of this project composition being purely software applications, the list of hazardous materials directive is a necessary piece of knowledge to be mentally aware of with respect to environmental considerations.

B. Disassemly

The Stock Market Trader Bot is a product that is of swift and easy installation in any standard computer. Furthermore, because of the discussed cross platform constraint, the Bot will be operable in any operating system such as Macintosh, Windows, and Linux. The user will be in full control of whether or not they wish to have the Bot be in active performance at all times, or for the Bot to be powered off and of no specific functionality. Moreover, the Bot is software based and uses pythonic code that is available cross platform amongst all standard computers. However, if an issue, or bug, is brought to the team's attention, a necessary software update will be constructed to promote constant improvement and swift utilization of the software. This means that the disassembly and reassembly of the Bot operation should be of no issue as a direct result of its cross-platform ability along with the easily configurable user interface.

C. Hannover Principles

The Hannover Principles consist of statements, or fundamentals, utilized in design phases of new products and projects with the consideration of the specific environmental effects. As outlined by William McDonough and Michael Braungart, some of these fundamentals are listed as follows:

- Recognize interdependence
- Respect relationship between spiritual and matter
- Accept responsibility for consequences of design decisions
- Create objects that safe and promote long-term value
- Eliminate waste
- Understand the limitations of the design
- Seek constant improvement

The team will adhere specifically to the constant pursuit of improvement through constant software updates, eliminate waste by focusing on software development, understand the limitations of the design by comprehending that a Bot cannot be correct in its predictions 100% of cases, and recognize the interdependence of a profitable Stock Trader Bot along with portfolio breadth (available monetary investment) and portfolio composition such as the specific stocks to be invested in and their history of profitability. In conclusion, these Hannover Principles are a well-constructed set of statements that will be followed for the designing of the Stock Market Trader Bot because of its technically sound fundamentals.

D. Life Cycle Impact Assessment (LCIA)

The Life Cycle Impact Assessment is an engineering technique that is utilized for environmental awareness of the materials being utilized in the implementation phase of a project. This technique involves assessing the impact the Trader Bot will have in times of repairment, performing maintenance, and distribution. With respect to times of repairment and maintenance because our product is software formulated, repairment will be easy with respect to updating the program of any bugs that are present. This means, our LCIA is of high quality and swiftly performed because of our project being software based.

In conclusion, the environmental considerations are of high importance for our Stock Market Trader Bot, regardless of its technically dominant nature. The RoHS is important to consider for all electrical and computer engineering product, the easy disassembly of the Trader Bot through its cross platform functionality allows for it be widespread amongst all standard computers, the Hannover Principles provides environmentally considerable fundamentals for designing of the Bot, and the LCIA provides an appropriate environmental assessment of the repairment and maintenance of the Bot through software and programming updates to improve overall the performance of the Bot.

XIV. SUSTAINABILITY CONSIDERATIONS

Sustainability considerations are imperative to creating a product that will uphold its value and even increase it in the long term. Creating a non-sustainable product would reduce the value and confidence of the product thus hindering its potential and long-term growth opportunity. The environment also plays a large factor when developing a product and the designers must take it into consideration. To create a product that negatively affects the environment is detrimental to society and should be reconsidered as a whole.

A sustainable design will use renewable resources and ensure the environment is not affected or positively affected by its outcome. To accomplish a sustainable design that will be long-lasting and not affect the environment in a negative way, the team has decided to focus on a pure software-related design of the product. This way the environment be minimally affected by the design outcome and the team can primarily focus on developing code that is long lasting, clean, and maintainable to ensure future updates are easily added.

A. Software

Sustainable, readable, long-lasting software is achievable especially when using a language such as python which looks very much like the English language. Writing sustainable code consists of writing clean code. Writing clean code is imperative because clean code is easily understood by other developers who may come along to read or change it. A few methods of writing clean code consist of "using proper variable names, writing solid functions, commenting on code, and providing documentation of the code as well" [43].

Using proper variable names allows for increased readability within the code thus leading the reader or editor to be able to read and maintain or change the code easily. Pertaining to commenting on code, this is important for the readability and maintainability of the code by another user. Sometimes even if code is written clearly with proper variable and functions, it could still be many lines of code that can be hard to understand at first. Comments throughout code help decipher what variables and functions are doing and assists future users in determining how to make changes necessary for maintenance or additions of new features. Code documentation has a similar role to comments but are much more in-depth and can explain why and how functions behave. Documentation is the quickest and best reference for developers to begin to understand code and is imperative for having future users understand a given codebase.

Solid functions should have "lesser arguments, be small, do only one thing, be D.R.Y and have no nested control structure" [43]. By following these rules for writing better functions, our outputs and the readability of functions can be deemed as proper and will be easier to maintain in the long-term thus affecting its overall sustainability. Functions that will be used in the codebase should have a small number of arguments. This makes the function more readable and maintainable in the future. The smaller the function is, the better as well. This because small functions are easier to maintain and to change later on in the development cycle. We can achieve smaller functions by using the D.R.Y (Do not repeat yourself) principals which explains that creating code that is redundant can always be reduced and is unnecessary to a codebase.

B. Life Cycle Assessment

This technique assesses how environmental factors and aspects could be affected by the project over its life cycle. It aims to describe the projects potential impacts on the environment and how it could then affect society. The LCA (Life Cycle Assessment) helps the team gather information about services and effects the project will have on the environment while covering the relevant energy and required inputs by the team's software. This technique will also provide an interpreted

form of results that will aid in the decision making of the project, providing the team with a more accurate and confident decision-making process.

The four steps of the LCA consists of a goal and scope definition, inventory analysis, impact assessment, and interpretation. “The LCA is also an iterative methodology, so the project will be continuously refined as it is developed” [44]. For example, if our project the stock trader bot was to run out of data or require more historical stock data, the LCA will provide a means to revise our goal and scope. Following the LCA also makes the product more sustainable in the long run through the utilization of the four steps of LCA mentioned earlier. The first step, to have a goal and scope definition ensures that the project work is performed consistently. This is done by “first performing an analysis on the product followed by a simplification of its reality” [44]. Doing this simplification allows the process to be broken down step by step and can be translated into goals for the project. The inventory analysis step allows the team to look at the environmental inputs and outputs associated with the product. Our team purely teals with software for our product, so this step will not be taken into consideration. Life cycle impact assessment occurs when drawing conclusions that allows the team to make better decisions through realization of how the project will affect the environment. For the final step, interpretation, the team will check whether or not the conclusions are well-substantiated by complying with the ISO 14044 standard [44].

By complying with these sustainability standards, the team will be able to create a long-lasting product that is also efficient. The impact our project will have on the environment and society will be taken into consideration using the software and LCA standards to provide the team with a guideline during the design phase. The overall goal for the stock trader bot is to be as sustainable as possible. This involves writing clean, bug-free code that is also clear and easily maintainable. We also take into consideration the goals and outputs of the product and the effects it will have on society using the LCA. Our product will aim to have the least amount of impact on the environment by being a fully software-based product that is also maintainable and sustainable through being elegantly written.

XV. MANUFACTURABILITY CONSIDERATIONS

Designing for Manufacturing consists of a practice that highlights products that are easy to manufacture and have a lower cost while being manufactured. By adopting the DFM, the team will be able to address issues with an affordable solution for the project. Saving money and time are vital to a products success especially in the long-term. The team must also be aware that making correct design choices in the early stages of the project will make the process easier and cheaper in the long run. Failure to comply with certain manufacturability considerations could cause the project to become too complicated or cost too much in the future. To produce manufacturability principles within our design the team will adhere to the following design principles when designing the stock trader bot:

1. *Simple Design and reduced number of parts*
2. *Design for Testability*
3. *Design for automated production*

Principles mentioned above are chosen with relevance to the teams project the stock trader bot. A simple design will be followed with as little parts used as possible, as well as being easily testable in an easy-to-use environment and finally being automated to produce results desired by the users of the bot. These principals will aid in developing the manufacturability of the bot especially in its early stages.

A. Simple Design and reduced number of parts

When it comes to the design of the stock trader bot, the team will use as few parts as possible by default since the bot will be purely software based. There will be no necessary hardware and will only take a personal computer to run the program as well all being the only requirement to run the bot. In terms of a simple design, the bot will by written in an easily readable programming language, python. It will also use the functional programming paradigm which contains a repository of functions that is created to be tested separately thereafter. These functions will be written tightly with as few arguments as possible and be made to be readable by other developers who want to access them. Reducing the complexity of the codebase is essential to having a simple design for the bot and can be achieved by using the paradigm and testing strategies as mentioned above.

B. Design for Testability

Testing the bot will be simple due to a planned implemented UI feature in which users can perform stock trades or let the bot handle the trading for them. The functional aspect of the bot will also be imperative to implement since unit testing becomes an easier process. Unit testing consists of testing each function of the bot to determine when it will work and when it will fail. Writing failures for these unit tests is essential to determine if the functions are working properly. Confounding factors have yet to be taken into consideration, so making the bot fully testable is essential to prepare for those situations. This can be done through code maintenance and debugging code to verify whether or not the bot is operating as intended.

*C. **Design for automated production***

Design of the bot is intended for stock trading and stock evaluation to be entirely automated. This principle is imperative to the team's project since producing the software consists of an automated process in itself. To create the bot, it should be as simple as to download necessary libraries required in python, use a python IDE and to then run the program. Creating the product based on the design for automated manufacturing (DFAM) will set the team up with success down the line when pertaining to the automation of producing several bots in the case of widespread use.

Manufacturability is imperative to a projects design. Keeping the project low cost and affordable while being aware that the first stages of the project are imperative will make the later stages of the project easier to maneuver through. Having a simple design with a minimal number of parts allows the stock trader bot to be easily re programmable and can be produced quickly. The bot will also be designed for testability, implementing UI features which users can perform trades on or allow the bot to handle the trading for them. Functions of the bot can also be tested easily because of a functional approach that was taken when programming the bot. For automated production, the bot will be simple to be reproduced since it will be contained in a git repository. Reproducing the bot should be as simple as to have the necessary libraries downloaded as well as cloning the project to a user's own computer. With these manufacturability considerations, our design will be led into a successful outcome in the early and long-term phases of the project.

XVI. ETHICAL CONSIDERATIONS & SOCIAL IMPACT

Considering ethics and social impact is imperative to any engineering project especially in a modern technological era. This is due to the fact that as technology grows more impactful upon society, so does its repercussions. To ensure the project is not infringing on anyone's rights, the designers must provide their own vision of the good life by considering these issues. Labeling and pondering these ethical dilemmas can become a complex process but is necessary to ensure designers and engineers comply with the IEEE Code of Ethics. The code of ethics is essential for all engineers to adhere to and if the project does not comply with this code of ethics, there will be an ethical dilemma which can then be addressed by using the Ethical Model Theory.

A. Ethical Considerations

The stock market trading bot is initially going to be designed for local use and with its growth and success, will eventually be globalized. An analysis conducted by the team regarding the ethics of the project will be developed. Determining the ethical and societal impact the project can have on its users will be due to adherence in following the IEEE Code of Ethics. As a team, ensuring the safety and ethics of the project is an upmost priority which will be conducted by following the codes described in the IEEE Code of Ethics:

1. *To hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, and to disclose promptly factors that might endanger the public or the environment;*
2. *To avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist;*
3. *To be honest and realistic in stating claims or estimates based on available data;*
4. *To reject bribery in all its forms;*
5. *To improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems;*
6. *To maintain and improve the technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;*
7. *To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;*
8. *To treat fairly all persons and to not engage in acts of discrimination based on race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression;*
9. *To avoid injuring others, their property, reputation, or employment by false or malicious action;*
10. *To assist colleagues and co-workers in their professional development and to support them in following this code of ethics.*

With the Code of Ethics brought to light, the design of the project and ethics following it will be in constant compliance with the ethical standards for all engineers. Prioritizing the health and safety of the users and the general public will always be a consideration and a top priority in fixing if any codes were to be broken. Not adhering to the codes can result in negative legal and moral implications which can cause future complications. By following the Code of Ethics, we are able to determine an ethical dilemma that could violate a code.

Suppose after following the Code of Ethics and ethical design principals the bot becomes fully functional. A user is then introduced to the bot and expects to make at least a 10% ROI from learning how to use the bot. But after a week of trading with the bot, they do not reach the expected 10% ROI. Instead, they make around 5% or break even. This would break a Code of Ethics by not being honest and realistic with how the bot produces results. The user may feel like they have been misinformed or lied to. We will now use the Ethical Theory Model to figure out what the best possible solution to fix this issue would be.

By utilizing four ethical theories, we will determine which solution is best. The following theories the team will be applying are:

- Utilitarianism: What will generate the highest advantage or least amount of harm for the largest number of people.
- Ethical Egoism: A morally correct action that promotes and values the interest of the team only
- Kantian Ethics: The option to willingly decide, rule, or policy that the team will follow in case of a future or similar issue.
- Rights Ethics: A morally correct action that respects the rights of individuals within a society by following those ethical considerations pertaining to the stock market bot and its active trader users.

TABLE XXI. Solutions for Ethical Dilemmas

Options	Description
1	Deny the existence of the issue
2	Issue a warning to the customer that the bot may not produce the expected 60% success rate
3	Provide an update to the issue to provide users with at least a 10% ROI
4	Only provide an update if there is long term evidence of the system producing less than a 10% ROI

TABLE XXII. Weight of Ethical Theories

Option	Utilitarian	Egoism	Kantian	Rights	Score
1	0.00	0.75	0.00	0.00	0.75
2	0.50	0.50	0.75	0.25	2.00
3	1.00	1.00	0.75	1.00	3.75
4	1.00	1.00	0.25	1.00	3.25

According to Table XXII, the best option for dealing with the presented ethical dilemma would be option 3. Option 4 is also a strong contender and is practical but providing an update itself would ensure that the system does not act in a way which does not produce results according to how it was designed and promised. Option 3 will solve the ethical dilemma, updating the software to provide users with the 10% ROI should be a priority if this dilemma were to ever present itself.

To conclude this section, working with trading exchanges and the stock market brings about sensitive ethical issues with respect to finances. Because of this, the team had to outline our response in respect to the dilemma and how the four ethical theories are going to consistently ensure that ethics and morals of this product are maintained. Prioritizing the health and safety of the users and the general public will always be a consideration and a top priority in fixing if any codes were to be broken.

B. Social Impacts

This section illustrates how the stock trading bot will contribute to society, locally or globally. The initial intent of the stock trading bot is to be usable on any team member's local system. The overall goal of the bot is to produce a positive ROI based on trading algorithms, hedge fund tactics, and past market data to make future stock price predictions. If the bot were to be a major success, it would be produced and sold to hedge funds or private investors to improve their trading abilities and add another essential tool to their repertoire.

Initial intentions of the design are to be utilized in a local system by the team members and mentor. Ideally, this provides a safe and alternate method to manual trading by applying the python language to use algorithms to automate the process of trading stocks. The bot will perform trades on its own set by the user's parameters providing a convenient method of stock trading. The vision for the good life is to have a stock trading bot that is functional, accurate, and easy to use for users. We also envision a future in which the bot can become commercially successful, eventually becoming a convenience and necessity for novice and professional traders alike.

In regard to ethical considerations and social impacts, the team must make sure there are no ethical dilemmas that breaks the code of ethics. If a dilemma were to surface, it will be addressed using the Ethical Theory model in order to uphold the ethical standards and to create a product that will improve the lives of those who use it. Analysis of possible and occurring dilemmas is imperative to the success of the product. Without analysis of ethical dilemmas, the product may falter or stray from the path of the vision the team has for the good life.

XVII. CONCEPT DEVELOPMENT

The overall goal of this section is to walk readers through how we plan to implement our product phases to provide a clear image on how we aim our final concept to be through data inputs, programming, machine learning models, and a friendly user interface. This section will illustrate multiple options for creating the project and consider their advantages and disadvantages. This way, the team can determine the optimal solution for the project. One way of describing the project is through a concept fan where the Stock Market Trader bot will be split into several components that must be considered: the data input, programming language, machine learning model, and how the project will output its data. A concept fan describing the previous is shown below.

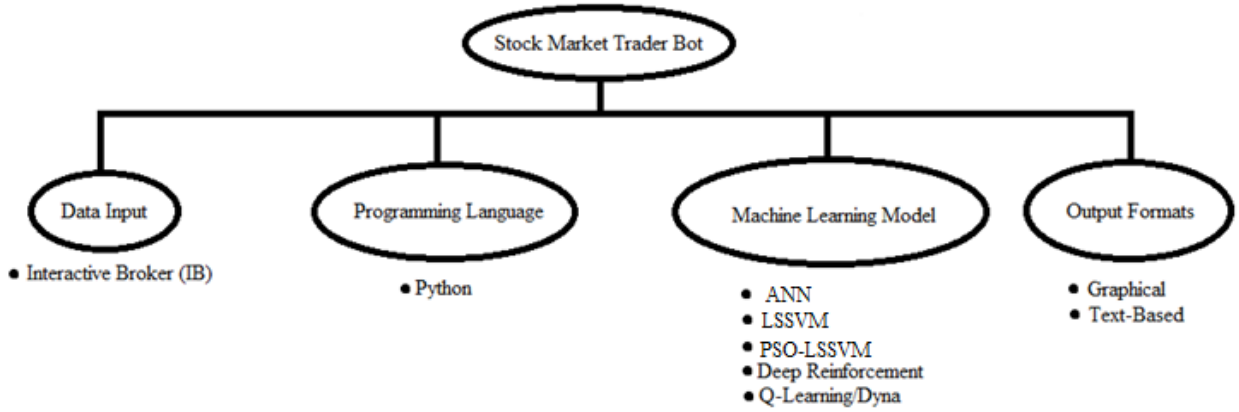


Fig. 24 Project Concept Fan

1) Data Input

Taking in data is a core part of the algorithm we are developing. While we could have the user input stock data, this will not satisfy objective #4, which is User-Friendly. Instead, we could have pre-loaded data into the program. However, the data will need to be updated as there is the risk the program can buy a stock that is no longer available. Therefore, to accomplish objective #4 and to have up-to-date data, we have decided to do this through an Interactive Broker (IB). In the book, *Algorithmic trading with interactive brokers: Python and C++* by Matthew Scarpino, Chapter 6 describes a method for obtaining data from IB's servers. To summarize, the application will connect to a Trader Workstation (TWS) through the TWS API through a socket. Then, the TWS works with the IB server to get the information back to the client. This information is the stock data we wish to input into the algorithm. This way, the algorithm can obtain data kept up to date by the IB. [25]

In summary, using the TWS API to connect to IB servers give the following advantages and disadvantages over user-given data and pre-loaded data:

a) Advantages

- Aims to primarily satisfies objective #4, being user-friendly. This is essential because it is our duty to fulfill the objectives of this product.
- Provides more recent data for more accurate results
- Well-Documented method for using TWS API

b) Disadvantages

- Dependent on third party's systems being available

2) Programming Language

Once the data is in place, the algorithm will need to be made. The team has decided on python to be the programming language; this is because python has an extensive list of libraries that focus on machine learning and mathematics and has already been established as a go-to programming language for many machine learning applications. Some of these libraries are known as Pandas, NumPy, and Scikit-learn. These well-established libraries allow us to handle the data efficiently and save the team the time it would take to write the libraries in another language. In addition, Python code tends to be more human-readable than other programming languages such as Java and C. However, having various libraries and being human-readable do come with consequences. Python is a high-level language, making it slower than low-level languages. Being able to crunch massive amounts of mathematical functions and data manipulation methods also leads Python to be resource-intensive, in addition to the higher power consumption. Below is a summary of the advantages and disadvantages that Python provides:

a) Advantages

- Well-documented performance in machine learning
- Various libraries focusing on machine learning, mathematical functions, and data manipulation
- More human-readable than most programming languages
- Efficient mathematical operations and data manipulation functions (Pandas/NumPy)

b) Disadvantages

- Slower than many other languages

3) Machine Learning Model

Once we have the data input and programming language decided, we will need to determine what machine learning models to use. Many different models are available, but it is essential to pick an optimal model for a specific purpose. In the team's case, the goal is to create an algorithm that can analyze stock data and predict future trends using the previous data. To train the algorithm most optimally, we will consider multiple machine learning models:

a) ANN [26]:

The Artificial Neural network, or ANN, is a machine learning model implemented into stock trades. The article, however, notes that it suffers "from the over-fitting problem due to the large number of parameters to fix, and the little prior user knowledge about the relevance of the inputs…" [26]. Therefore, ANN would not be a good fit for the project, as we want the results to be as accurate as possible and user-friendly. In summary:

- Advantages:
    - History with being used in stock market trading
- Disadvantages:
    - Over-fitting problem requiring constant maintenance
    - Does not fit with objective #4, User-Friendly

b) LSSVM [26]:

The Least-squares Support Vector Machines (LSSVM or LS-SVM) is a reformulation of the Support Vector Machines that were made to avoid the over-fitting and other limitations of the ANN model. According to [26], ". LS-SVM uses a regularized least-squares function with equality constraints, leading to a linear system which meets the Karush-Kuhn-Tucker (KKT) conditions for obtaining an optimal solution.". The text mentions the need to select an appropriate method for selecting the LSSVM's free parameters to make the LSSVM robust. A summary of LSSVM's usefulness is below:

- Advantages:
  - Avoids over-fitting issue that occurred in ANN
  - Variation of a well-made model
- Disadvantages:
  - Requires a method to keep the LSSVM robust and optimal

c) PSO-LSSVM [26]:

This model combines a search method and a machine learning model: Particle Swarm Optimization (PSO) and LSSVM. In the journal article [26], this machine learning model optimizes the LSSVM with the PSO algorithm. The summary of the PSO-LSSVM model is shown below:

- Advantages:
  - More accurate than ANN and LSSVM in stock trading simulations
  - Uses PSO as the method to keep LSSVM robust and optimal
- Disadvantages:
  - Requires implementation of two complex concepts

The graphs provided by [26] also have more accurate results than the other two machine learning models. This was supported by the Mean Square Error (MSE) provided by [26] shown below,

TABLE XXIII. MEAN SQUARE ERROR FROM [26]

Algorithm / Company	PSO-LS-SVM	LS-SVM	NN-BP
Adobe	0.5317	0.5703	0.8982
Oracle	0.6314	0.8829	0.9124
HP	0.7725	1.2537	1.9812
American Express	0.7905	1.0663	2.8436
Bank of New york	0.4839	1.2769	1.9438
Coca-Cola	0.6823	0.9762	1.7975
HoneyWell	0.9574	1.3371	2.1853
Hospera	0.8694	0.9320	1.4640
Life Tech.	0.7713	1.3221	1.3492
Exxon-Mobile	1.1000	1.6935	2.4891
AT & T	0.2911	0.4673	0.4055
FMC Corp.	1.5881	2.1034	3.5049
Duke Energy	0.1735	0.6097	0.6010

MSE Calculated across various stocks [26]

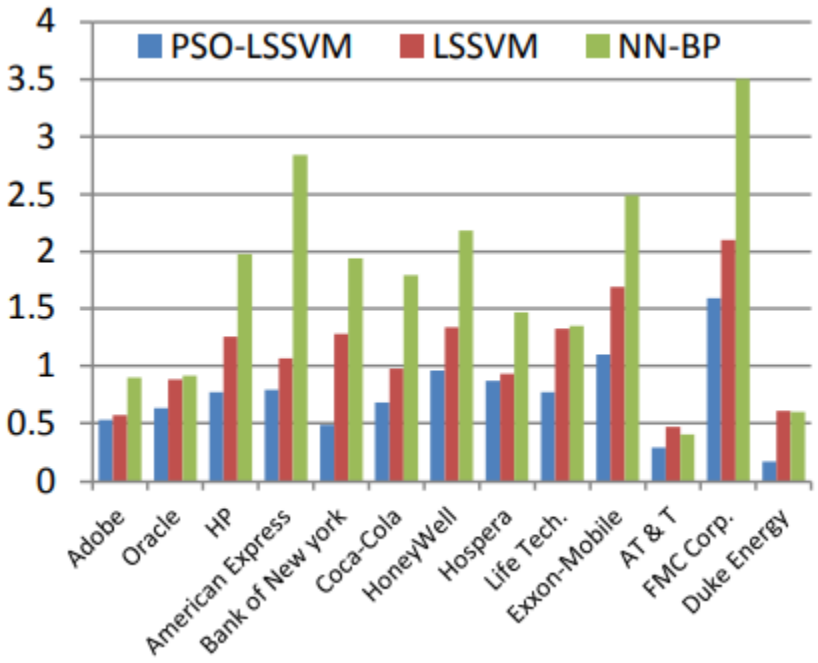


Fig. 25 Bar Graph View of MSE Results Across Various Stocks

d) Deep Reinforcement [27]

Deep Reinforcement Learning is a combination of ANN along with a reinforcement learning algorithm. The algorithm guides the ANN to the solution by "rewarding" the ANN when desired actions are taken. [28] The article [27] uses a sentiment analysis model and a recurrent convolutional neural network to predict stock trends through financial news. Ultimately, the authors determined that the Deep Reinforcement Learning model can trade stocks. However, it is also acknowledged that reinforcement learning models are not the best way to trade stocks, stating that "Such non-deterministic problems can only be solved with neural networks" [27]. As a side note, the authors mention that the architecture used can be scaled to do multiple stocks through a master network controlling numerous other networks. While we could potentially exploit this model, the paper using it, [27], uses a different data input:

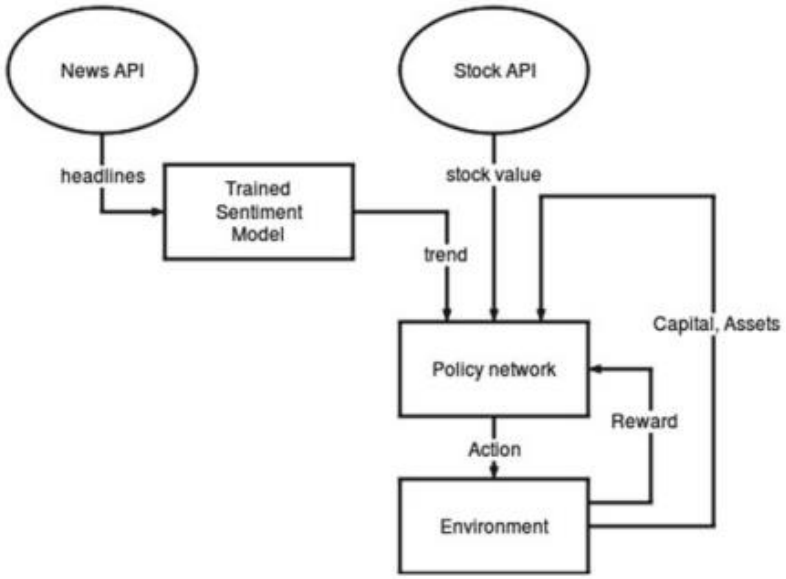


Fig. 26 The Architecture of the Proposed Deep Reinforcement Learning for Trading Stocks [27]

As can be seen, the model uses News headlines to predict stock values instead of the stock data itself. The summary of the model's potential contribution to the project is listed below:

- Advantages
  - Potential for different data inputs
  - Shown to have potential in trading stocks
- Disadvantages
  - Deviates data input from stocks to news headlines
  - Not confirmed to be scalable, only potentially scalable
  - Noted not to be the best solution for stock prediction

e) Q-Learning/Dyna [29]

Q-Learning and Dyna are alternatives suggested for stock trading in the Udacity classroom [29]. Q-Learning is described as model-free; instead of conventional T (transitions) or R (rewards), it uses Q-Values or Utility Values generated as it interacts with the world. Doing this allows Q-Learning to arrive at optimal solutions throughout many iterations. However, it was noted that it would require many iterations to create a profit consistently and may cost a lot of money to train this model. Below is a summary of Q-Learning:

- Advantages
  - Suggested for Stock Trading
  - Suitable for the complexity of Stock Trading
- Disadvantages
  - Costs money to achieve good results
  - Needs paper-trail mechanism

Dyna is another model made in response to Q-Learnings learning costliness by Rich Sutton. Dyna uses T (transition matrix) and R (reward Matrix) models. Dyna also incorporates Q-Learning, but Dyna can simulate hundreds of simulations in addition to a real-life run in the stock market. Dyna will run a real simulation and then will run many non-real simulations, which costs less to do than Q-Learning itself would do.

- Advantages
  - Less costly than Q-Learning while still converging (finding an optimal solution)
  - Suitable for the complexity of Stock Trading
- Disadvantages
  - Seemingly, the best solution may get exploited in massive amounts and rendered ineffective later

4) Output Format

The Stock Trader Bot ultimately will buy stocks on its own or suggest to the user what stocks to buy/sell. Showing the user what happened, or what stocks they should buy, can take many forms. Below are two options; however, it may be best to represent the output in both to the user to align with objective #4, User-Friendly

a) Graphical

A Graphical Representation showing points where the Stock Trader Bot will buy/sell or where it thinks the user should buy/sell can summarize all the information a user needs to know quickly and efficiently; however, it may not be easy to see these points exactly. In summary:

- Advantages
  - Easy Visual Summary
  - Shows multitudes of information at once
- Disadvantages
  - May not show specifics

b) Text-Based

A Text-Based Representation may prove helpful in providing specific information to the user. However, it can become long and unreasonable to read if the output tries to show multiple details simultaneously. In summary:

- Advantages
  - Can show more specific information easily
- Disadvantages
  - Cannot represent multitudes of information at once reasonably

Both tables XXIV and XXV effectively draw correlation of the objectives prioritized in the concept development for the product through a weight calculation table alongside a concept selection table for best practice.

TABLE XXIV. WEIGHT CALCULATION TABLE

	Safety	Risk	Security	Cross-Platform	Accurate	Profitable	Modular	Auto & Manual	User-Friendly	Geometric Mean	Weighted Mean
Safety	1	1	1/3	5	1/7	1/7	5	9	5	1.25	0.10
Risk	1	1	5	7	1	1	3	9	5	2.56	0.20
Security	3	1/5	1	7	1/3	1/3	5	9	5	1.67	0.13
Cross-Platform	1/5	1/7	1/7	1	1/7	1/7	1	1	1/3	0.38	0.03
Accurate	7	1	3	7	1	1	7	5	5	3.09	0.24
Profitable	7	1	3	7	1	1	7	1	5	2.58	0.20
Modular	1/5	1/3	1/5	1	1/7	1/7	1	1	1/5	0.33	0.03
Auto & Manual	1/9	1/9	1/9	1	1/5	1	1	1	1/5	0.33	0.03
User-Friendly	1/5	1/5	1/5	3	1/5	1/5	5	5	1	0.66	0.05
Total										12.85	

TABLE XXV. CONCEPT SELECTION TABLE

	ANN	PSO-LVSSM	Q/Dyna	
Safety	1/3	5	5	0.10
Risk	1/3	5	5	0.20
Security	1	1	1	0.13
Cross-Platform	1	1	1	0.03
Accurate	1/3	5	5	0.24
Profitable	1/3	5	5	0.20
Modular	5	3	5	0.03
Auto & Manual	1	1	1	0.03
User-Friendly	1	1	1	0.05
TOTAL	0.629	4	4.09	

XVIII. End Product Description and Other Deliverables

Through the delineation process and groundwork construction of an engineering product, it is pivotal to produce a coherent, logical end product description. The End Product Description and other deliverables explains to participating engineers and designers how the Stock Market Trader Bot will operate along with the functions it will perform. This section will be split into four subsections: End Product Description, Functions, Specifications, and Other Deliverables. For End Product Description, text and diagrams will aim to demonstrate the Bots final output. The Functions subsection will list the different software libraries created and used to formulate the algorithms of the Stock Market Trader Bot. The product also requires a list of specifications outlined in the subsection of this section that clearly states the items and technology such as inputs and outputs needed to operate the Stock Market Trader Bot. Overall, this section can be organized in a method based on levels. The End Product Description subsection is level 0 and is known as the black box, or algorithms and overview of input/outputs of the system. The Functions subsection is level 1 and effectively uncovers the functionality behind level 0. Lastly, level 2 will be specifications necessary for the functions to be operated appropriately.

A. End Product Description

Composing a concise, transparent end product description provides team members the ease of operation when performing the design and implementation of the product. Because the Stock Market Trader Bot aims to be fully modular as described in the objectives, this section will discuss the final production of how the client-friendly user interface will behave for both functional options of trading for the Bot: 1) Automatic Transaction Handline and 2) Client/User Intervention. Figure 27 is an extremely general depiction of how the operation of the Stock Trader Bot end product will be handled. Through this subsection, both functional operations and level 0 will be explained through text and images to piece together a better understanding of the end product description.

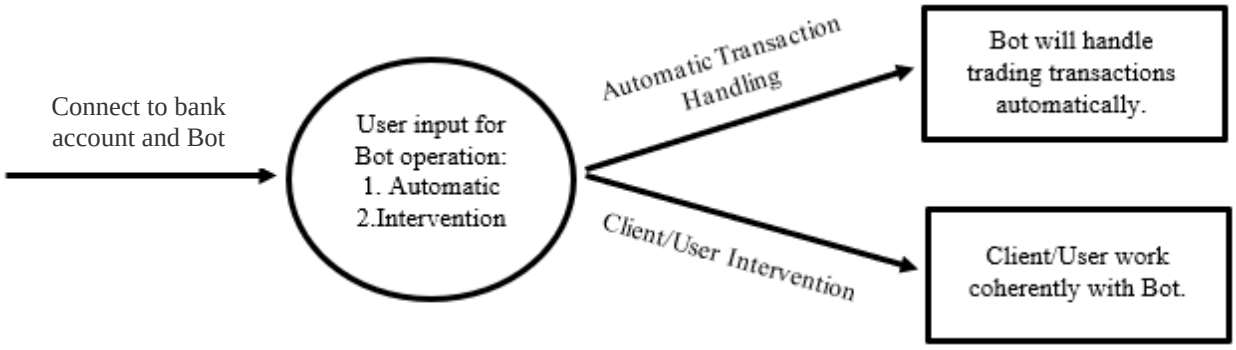


Fig. 27 End Product Description Overview

1) Automatic Transaction Handling

If the client is to select Automatic Transaction Handling as the contemporary operation, the Bot, in summary, will proceed to performing buy/sell orders of a predetermined portfolio of varying stocks based off algorithmically driven decision-making models. This specific option of operation for the Bot does not require any user intervention other than selecting a pool of stocks in which the user wishes to allow the Bot to manipulate through varying orders and exchanges. From here on forth, the Bot will use the library of machine learning algorithms it has been trained with to predict a stock's price based on historical data analysis ranging from approximately 25 years prior.

Furthermore, the Bot will also ensure to use economic theories to formulate a trend and moving average to understand the behavior of a select stock. If the product has been on market for less than 25 years, then it will utilize the data analysis from the day it entered the market. Lastly, as shown in figure 28, if the stock has been on the market for less than 5 years, then historical data analysis will work in conjunction with the Bollinger Bands that works with the standard deviation of the stock's short time on the market to decide when it is the most profitable time to buy/sell orders.

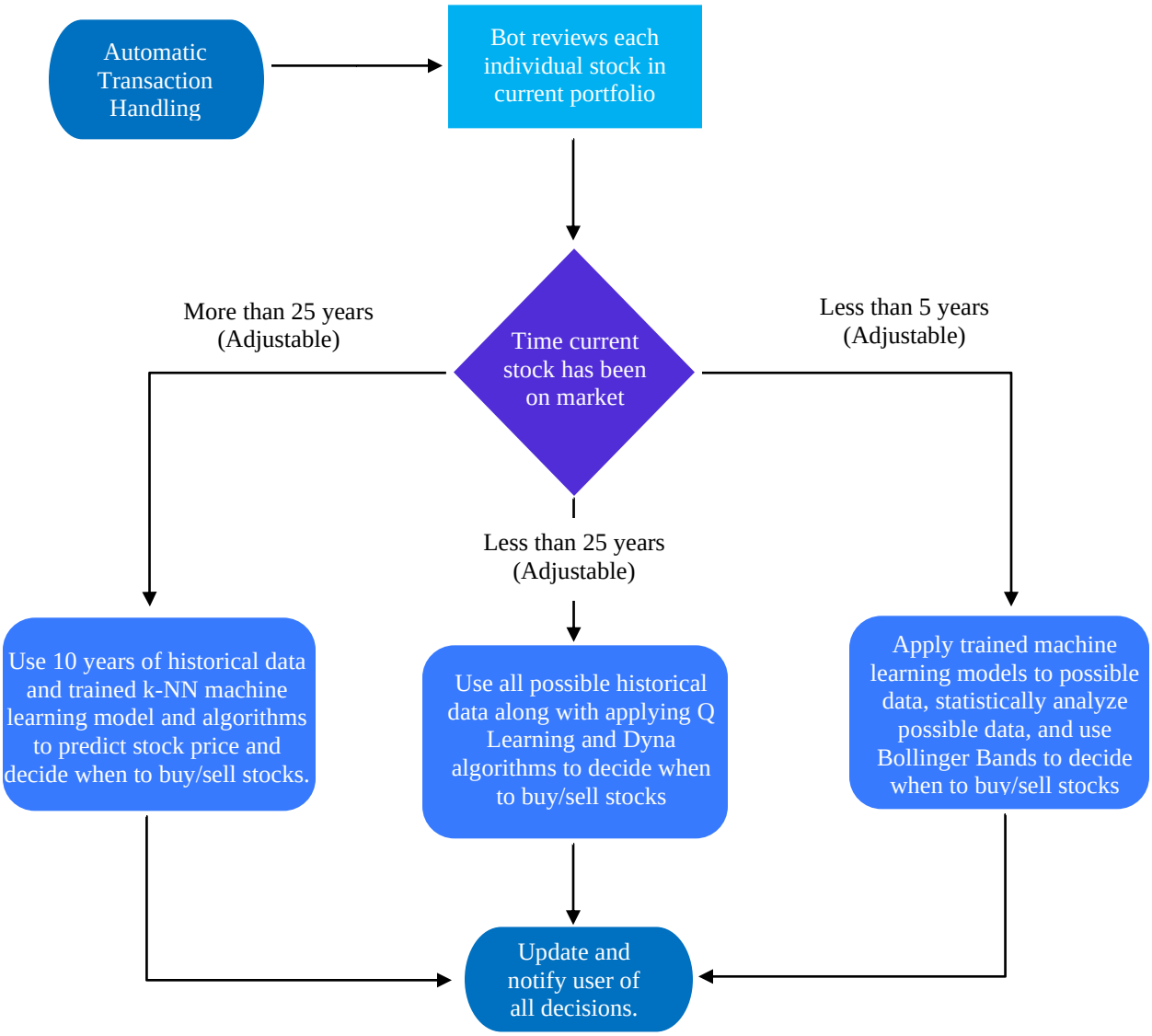


Fig. 28 General Overview of Automatic Transaction Handling

2) Client/User Intervention

If the client was to select the alternative option, "Client/User Intervention," the Bot continues to apply all its machine learning models, historical data analysis, Q Learning/Dyna algorithms, and economic theories. However, rather than automatically handling buy/sell orders and the decision-making process when controlling stock orders in a portfolio, this option will leave the choice to the user. In this functional option, the primary position and tasks of the system is to merely provide

suggestions and notifications to the user of when it recommends acting on an investment. As shown in figure 29, the only difference between the two functional options of the Bot is that the Client/User intervention option seeks to avoid acting on orders and serves to provide the user with insight into when the Bot believes there is an opportunity for a profitable change in portfolio.

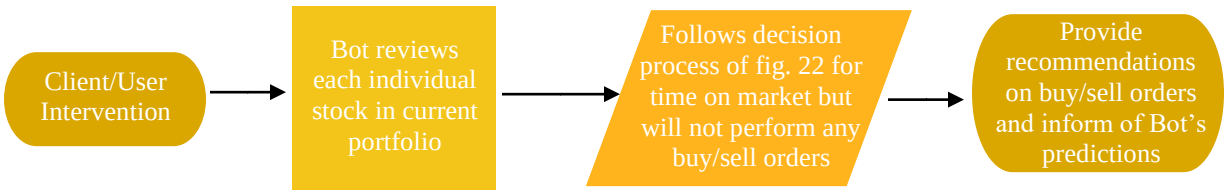


Fig. 29 General Overview of Client/User Intervention Handling

In conclusion, the End Product will have a clear user interface that allows for easy handling of the Bot's trading operations. Prior to any trading, the end product requires a clear connection to a proper bank account that will be established with the Bot to allow for legal and effective connection to ensure the ability to manipulate buy/sell orders. From this friendly interface, the client will be able to select whether it wants its current portfolio to be handled completely by the Bot using the Automatic Transaction Handling option or wants to work alongside the Bot using the Client/User Intervention option. In the automatic handling, the Bot will analyze each individual stock in the user's current portfolio and approach the buy/sell orders differently depending on its time in the market. After any decision has been made with respect to the user's portfolio, the system will clearly describe to the user what investment decision it made (buy/sell), why it made it (BB or algorithmic prediction), and the specific time it executed the order. This process of notification in the automatic handling option will be provided at the close time of every day the market is open. With respect to the Client/User Intervention option, the Bot will still follow the same process as the automatic handling with respect to analyzing a portfolio, however, it will hold back on executing buy/sell orders. Rather than buying and selling, the Bot will provide an immediate notification to the user of whether it recommends the user to act on an investment (buy/sell), why it recommends it (BB, prediction, or moving average), and when it recommends acting on investments.

B. Functions

In this subsection, the functions necessary to guarantee the end product description is accurate will be determined. As mentioned in the beginning of this section, the functions subsection accurately represents level 1 and will be organized to describe more detailed specifics as to the specific functionalities behind the end product description.

1) Data Science/Machine Learning

As mentioned in the previous subsection, regardless of a stock's time on market as well as the user's selected functionality option of the Bot, historical data analysis will be performed to analyze the behavior of individual stocks over an extended period of time. More specifically, the Bot will be trained to use data science principles to split the historical data into both a training and test samples. The training set will be compared to the test to measure the accuracy of the trained model. Over the time the stock has been on market, the Bot will plot the stock's value as individual points with respect to each day. At this point, once the historical data is plotted point by point, a machine

learning algorithm, k-nearest neighbor (k-NN), will be utilized to establish a relationship between the daily price of a stock depending on its specified number of neighboring points as valued in variable k [36]. Once this relationship has been established, this algorithm will be used to train the Bot model to create a connection as to whether or not the behavior overtime of a product indicated there was going to be an increase or decrease in the stock's closing price of the day. In turn, this will effectively provide a prediction of a stock's price for the following day. Finally, once the Bot is able to create a concrete prediction based on the training sample, it will be compared to a test sample and, according to the objectives, will be correct in its predictions 60% of the time to meet the profitability of 60% of investment decisions. In figure 30, a visual depiction is provided as to how the input and outputs of both of the Bot's functionality will generally behave. Following figure 30, table XXVI serves to indicate the input, output, and function of the data science and machine learning application of the Stock Market Trader Bot.

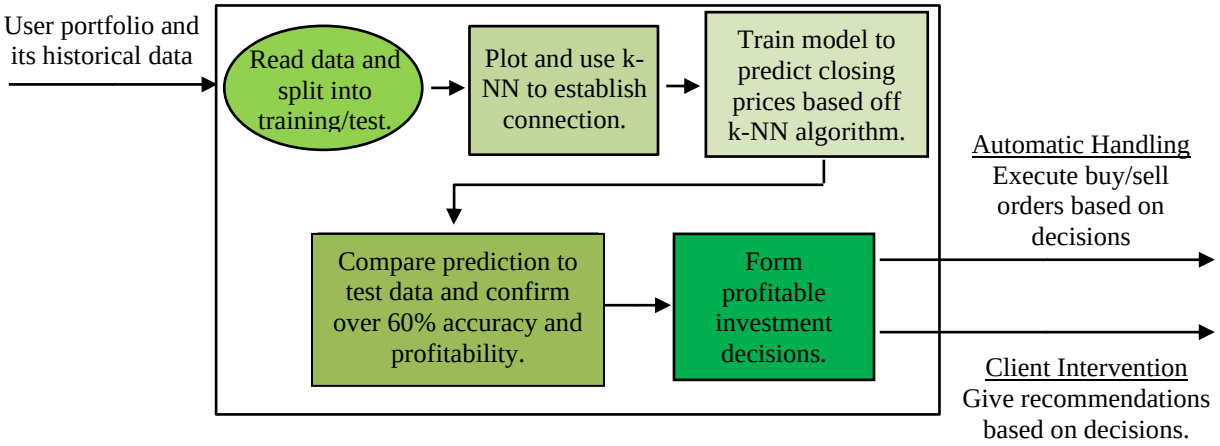


Fig. 30 Functionality of Data Science/Machine Learning in the Bot

TABLE XXVI. FUNCTIONALITY OF MACHINE LEARNING

	Automatic Transaction Handling	Client/User Intervention
Input	Predetermined user portfolio historical data.	Predetermined user portfolio historical data.
Output	Buy/sell orders based on predictions and trained k-NN machine learning algorithm.	Recommendations on investment decisions based on predictions and trained k-NN machine learning algorithm.
Function	Provide 60% overall profitability by using k-NN algorithms and economic theories to make investment decisions.	Provide recommended buy/sell orders based on trained algorithm and economic formulas.

2) Q-Learning/Dyna

As specified in the end product subsection, for stocks in a client's portfolio that has been on the market for less than 10 years, historical data analysis will be used coherently with Q-Learning and Dyna to promote profitable decisions in exchange orders. Q-Learning and Dyna are model-free reinforcement learning algorithms that are aimed to primarily find the best course of action, given the current state of the agent [37]. Using these algorithms will work well with the machine learning algorithms because it will produce a more intelligent Bot that better understands the market and stocks in a current portfolio. More specifically, these reinforcement learning algorithms serve to provide the Bot a higher percentage in prediction by formulating a trend amongst previously analyzed historical data. From this trend, the algorithm will begin to identify patterns in a stocks' behaviors and begin to make more accurate predictions based off these stock growth/decline tendencies it has taught itself over time. This is similar to the behavior of ad recommendation systems that are based on searches done by the user (e.g., user searches 'Samsung' and receives ads on Samsung products). In figure 31, inputs and outputs relative to the applications of these reinforcement models will be demonstrated in the form of a diagram along with Table XXVII that effectively illustrates the Q-Learning/Dyna function in a table.

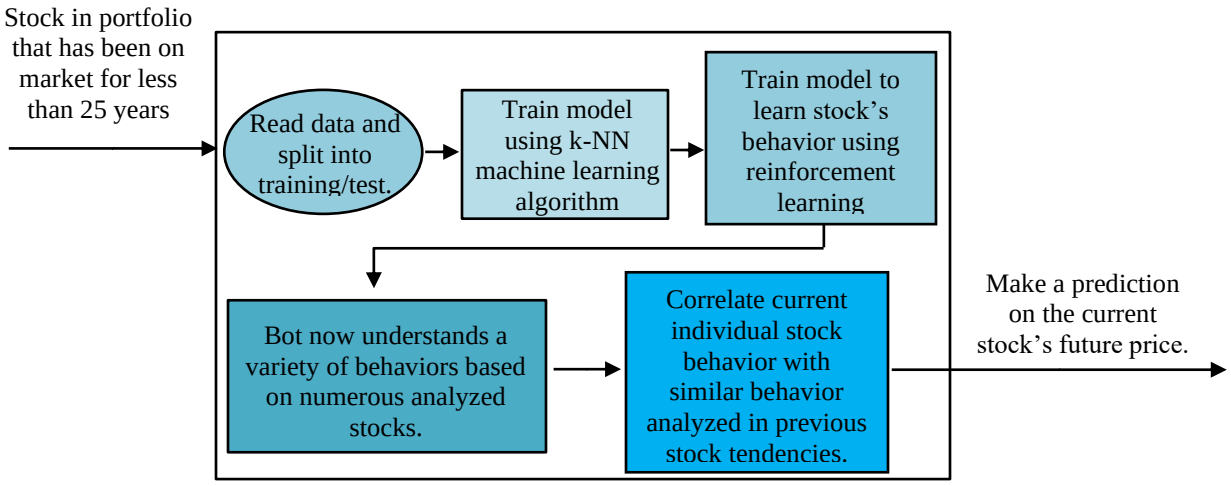


Fig. 31 General Overview of Reinforcement Learning Functionality

TABLE XXVII. FUNCTIONALITY OF REINFORCEMENT LEARNING ALGORITHMS

	Q-Learning/Dyna Algorithms
Input	Predictions based on historical data analysis.
Output	More sophisticated and accurate predictions for investment decisions.
Function	Algorithm will teach itself to make better predictions based on tendencies in previous decisions and learn to better analyze stock trends to be more accurate in predicting.

3) Bollinger Bands

As mentioned in the first subsection, when a stock has been on the market for less than 5 years, it will use Bollinger Bands (BB) along with the machine learning and reinforcement learning algorithms to ensure that proper surveillance is put on the stock's fluctuation. To summarize, BB are envelopes plotted at a standard deviation level both above and below a simple moving average of the price [38]. This works alongside the trained algorithms because the simple moving average is computed using k-NN machine learning while using a training sample to improve its own predictions through reinforcement learning. BB is especially important to use in fairly new stocks in the market because of the unexpected volatility and uncertainty in the behavior a stock may have as a result of its few times in the market and minimal historical data. Basically, if a stock price touches one of the BB envelopes either above or below, the Bot will decide as to what is the best move to make with respect to the user's current portfolio standings. For example, if a client has invested in a specific stock and the stock rises outside of the above envelope, falls back down, and touches the envelope on the way down, the Bot may identify that the price is relatively high and is plummeting. This will encourage the Bot to probably sell the stock at a higher price than normal to gain profit before it falls back down to the moving average. In figure 32, a general diagram represents how Bollinger Bands plays a role in the entire decision-making process as well as affecting the Bot's prediction for a stock with minimal historical data. Following figure 32, table XXVIII will serve as a table representing the inputs and outputs to perform the function described in this subsection.

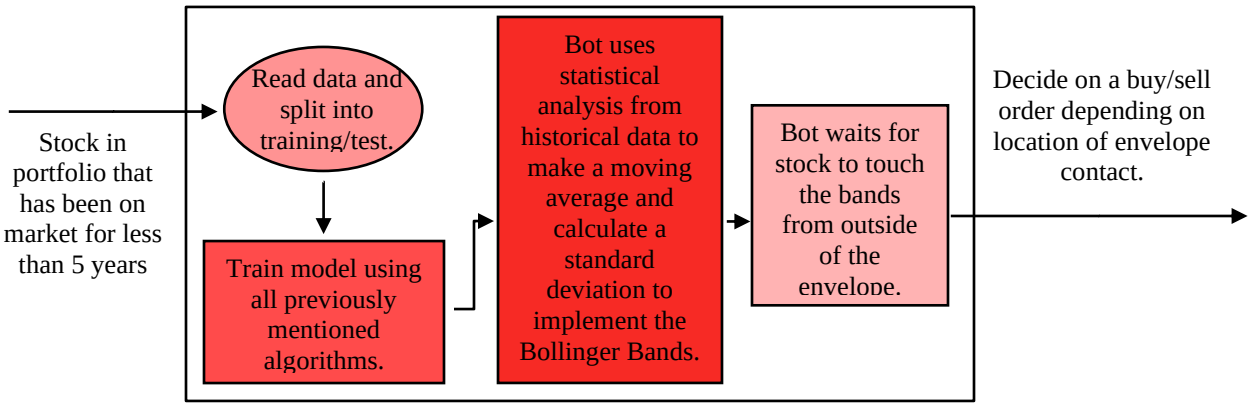


Fig. 32 General Overview of Bollinger Bands Functionality

TABLE XXVIII. FUNCTIONALITY OF BOLLINGER BANDS

	Bollinger Bands
Inputs	Statistical analysis of historical data.
Outputs	Signals to buy/sell orders depending on BB contact to counteract volatility loss.
Function	Use k-NN to compute moving average and statistical analysis to create BB a standard deviation above and below moving average to use as guideline for investment decisions.

C. Specifications

The Stock Market Trader Bot will provide novice traders the opportunity of a profitable experience in the stock market and trade at a high performing level with minimal user effort and activity, serving as level 2 of the end product deliverable. In order to allow for this opportunity, there are specifications with respect to the end product description that must be identified. Table XXIX provides a list of components that will be necessary to manipulate and run the Bot software and user interface.

TABLE XXIX. SPECIFICATIONS

	Components	
Bot UI Operation	User must select which functional option it wants the Bot to operate in	Connection to bank account
Historical Data	All individual stocks in current user portfolio	IB market data or Yahoo finance access
Buy/Sell Orders	Connection to a funded bank account	Trained machine learning algorithms and custom written data science library with functions by team members to make investment decisions.
Programming Language	Python for all machine learning and trading algorithms as well as the graphical user interface	Python to create a file that is compatible with friendly user interface to notify user of recommendations or decisions

D. Other Deliverables

This subsection illustrates deliverables such as information on the PowerPoint presentation, final report, user manual, and other addressable deliverables for the remaining time spent on implementing the Stock Market Trader Bot. Regarding the presentations and final report, viewers, and readers of these will have a clear depiction of all aspects pertaining to the design phase of the project. The presentation will serve to vocally and visually summarize the information presented in the final report as a way of attracting interest in the product. The final report will serve to officially document the entire blueprint of the Bot created in unity by the participating engineers. Lastly, the team members of this product will ensure to have a document that walks clients through the usage of the Bot and how to operate it through a user manual. This document will be in the form of a well-constructed word and pdf document that serves to ease the integration of the Bot into the trader's account as well as how to troubleshoot issues that may arise in order to allow for a smooth transition to automated trading.

In conclusion, the end product description and other deliverables is important to the overall outline of the product because it provides a clear understanding of how the designing and implementation of the Stock Market Trader Bot should proceed. Without a clear outline and description of how the final product is aimed to be operated, the creation of the product and objectives can be skewed, therefore, it is imperative to have an effective end product section that provides a concise, accurate model of the final product of the Stock Market Trader Bot. To summarize this section, a description of the constituent modules and their interrelationships was described in the end product description subsection, the functionality of the product and different modules was described in the function's subsection, the product specifications was clearly stated in the specification subsection, and the deliverables for next semester with respect to the Stock Market Trader Bot was evidently described in the other deliverables subsection.

XIX. PLAN OF ACTION

Making a plan that effectively distributes tasks among team members and the available time is essential to completing projects and efficiently using each member's time. Separating the entire project into parts and milestones will ensure that each part of the project is completed on time and helps visualize the end product. Separating the tasks in the plan for each member will ensure that each member completes an appropriate amount of work and holds them responsible for completing their work on time. Doing so keeps each member motivated to accomplish their goals and reach the milestones set by this section of the project. The plan of action will include a Statement of Work, which defines the project's scope, location, period, and deliverables and will aid in facilitating the distribution of time for the project. A Work Breakdown Structure will also be included to help effectively distribute time for each section of the project by indicating how work should be distributed relatively. Project Milestones will also be set to help keep the team motivated until the end of the project. The plan of action also splits the project into different phases, including the tasks needed to be done when each phase ends. By completing the final phase, we will also finish the entire project. This section will consist of a PERT chart to help visualize the end goal.

A. Statement of Work

The statement of work defines the scope, location, period, deliverables, and deliverable due dates to help facilitate envisioning the amount of work that has to be done, the method of collaboration that will be used for the project, the amount of time that the team has to complete the project, the tasks needing to be done to complete the project, and the due dates for the tasks of the project.

1) Scope

The Stock Market Trader Bot will be an algorithm capable of predicting future stock trades accurately. It will be accompanied by an application that provides a user-friendly GUI to allow the user options to modify the Stock Market Trader Bot. These options will be including a way for the user to choose between two modes, Automatic Transaction Handling, and User Intervention. By selecting Automatic Transaction Handling, the program will allow the algorithm to execute trades itself in a way to maximize profits for the user. User Intervention will enable the user to choose whether to execute a trade or not that the algorithm determined to be profitable. The algorithm will predict and execute trades to minimize risk for the user, implement machine learning, be entirely modular, and allow safe use for all users. The program will be cross-platform to maximize the number of potential users and securely execute trades.

2) Location

The location of the project's work is essential as all team members will need to agree on a location. This will facilitate team members' cooperation and ensure that each member knows how and where to get help from other team members. The team will develop the project and collaborate remotely, as the end product will be more software-based, allowing the team to work and collaborate from anywhere. Using version control software such as Git and GitHub will allow for this software and programming focused projected to be manipulated and coherently worked on remotely. Location is an essential consideration because the contemporary and following semesters are courses attended remotely, therefore, remote conditions must be set up accordingly.

3) Period

The project's period is essential to define as it shows how much time the team has to develop the product. Formulating a straightforward period allows for a precise timeline to be constructed to ensure the workload is widespread throughout the period. In addition, allowing the team to create a plan that effectively distributes tasks along the period and ensures that the project will be completed by the due date. This project will last the length of two university semesters each being approximately four months. More specifically, the duration of the projects work period is from May 2022 to December 2022.

4) Deliverables

- June 21, 2022 – First Proposal Due
- June 23, 2022 – First Presentation Due
- June 30, 2022 – Short Video Due
- July 21, 2022 – Second Proposal Due
- July 26, 2022 – Second Presentation Due
- July 29, 2022 – Project Demonstration Due
- July 29, 2022 – Final Proposal Due
- August 22, 2022 - December 3, 2022 – Implementation
- December 3, 2022 – Completed Project Due

*B. **Work Breakdown Structure***

The work breakdown structure defines how much work should be put into each project section. It will be compared to each other section of the project as a percentage and should add to 100% of the total. The work breakdown allows for proper time management to be applied and emphasize where time and effort should be prioritized. Sectioning the project as done in this subsection is extremely effective in creating a blueprint of the technical labor that will be induced by the implementation phase of the Stock Market Trader Bot. By sectioning the project, it will also help determine how much time will be put into each phase of the project, and a diagram of the work breakdown structure will help visualize how much work is left to do to complete the project. In the following page, an extremely clear and concise diagram will effectively demonstrate the Work Breakdown Structure unique to this product.

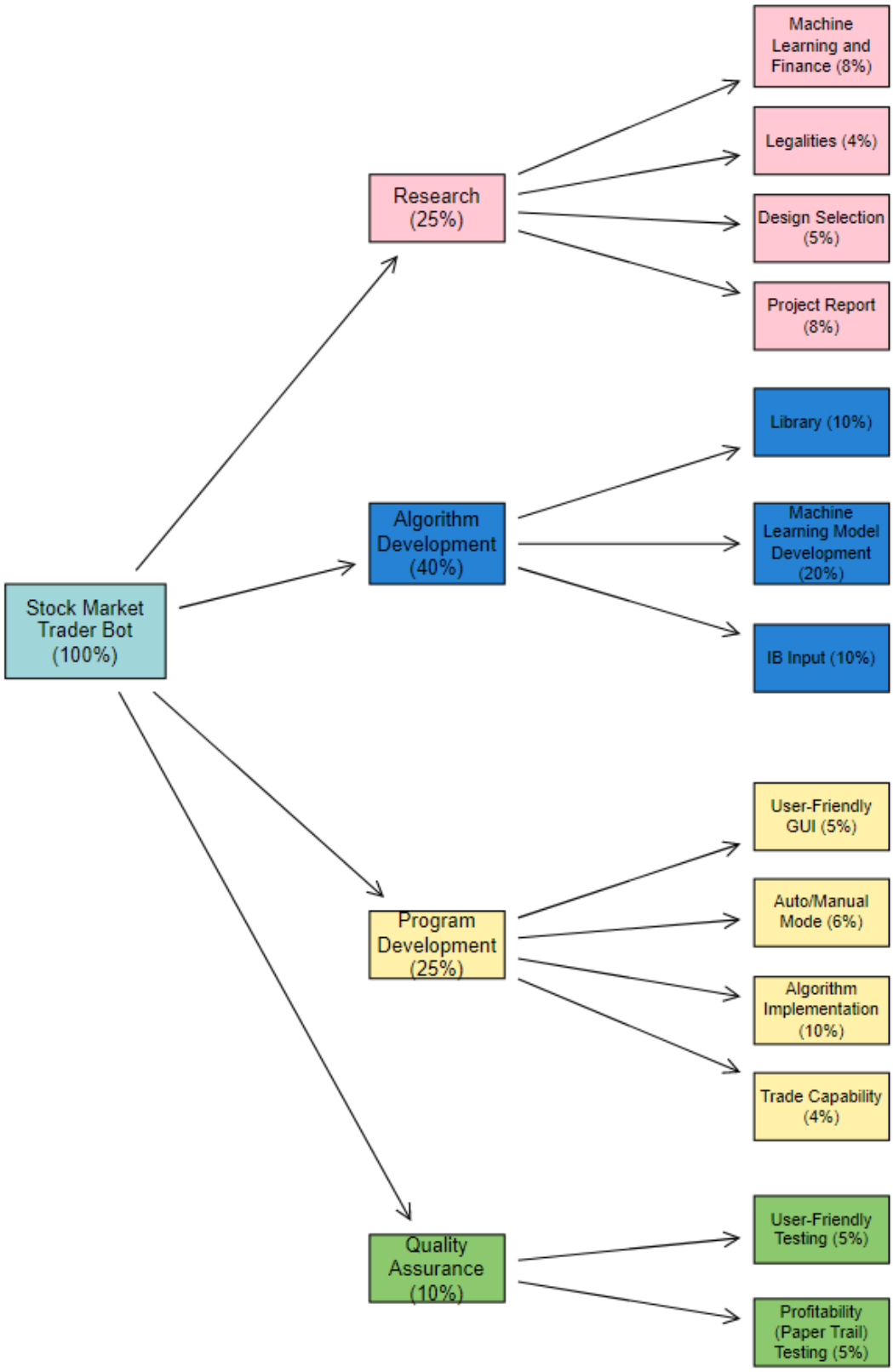


Fig. 33. Work Breakdown Structure

Fig. 33 shows the Work Breakdown Structure for this project; using this diagram, we will distribute the amount of time needed as well as the work for each task among each member to ensure we achieve the project deadline as well as make sure each member receives equal amounts of work to do. In addition, the diagram shows the amount of work left to do when each task is finished.

*C. **Phases***

The project will be split into phases, including the tasks needed to complete each phase and the deadlines for each phase. Doing this can motivate each member to complete each task and ensure the project is completed by the due date. The project will be split into 4 phases: Research, Algorithm Development, Program Development, and Quality Assurance.

1) Phase 1: Research

- Objective: Research the topic and compile several alternatives of implementations, then decide on a design and generate a report
- Approach: Using information surrounding the Stock Market and Artificial Intelligence available to the team and decide as a team on the design and generate a report
- Result: We will have a project proposal that outlines the objectives of the team, as well as a feasible design implementation of the project

The Research phase is intended to be completed early in the project's lifetime. Thus, it will start in May 2022 to July 2022. It is separated into Machine Learning and Finance, Legalities, Design Selection, and the Project Report.

2) Phase 2: Algorithm Development

- Objective: Develop a python library, as well as a machine learning algorithm capable of gaining IB stock information that can predict the future of stock prices accurately
- Approach: Using python, PANDAS, and NumPy, and guidance made available through the Udacity Course and python financial analysis books
- Result: Have an algorithm that is capable of predicting the future prices of stocks accurately

The Algorithm Development phase is the central part of the project and will require substantial development time. Therefore, it will span from June 2022 to October 2022. It will take the concepts selected in the Research phase and outlined in the report.

3) Phase 3: Program Development

- Objective: Developing a program with a GUI that is user-friendly, allows an auto/manual mode, implements the algorithm, and has the trading capability
- Approach: Using python, program a GUI and combine the algorithm and application as well as handle the stock trade exchange
- Result: An application that implements the algorithm and allows the user to make a profit

The Program Development phase will focus primarily on user-interaction and algorithm-stock exchange interaction. It will ultimately use the algorithm developed in Phase 2: Algorithm Development. Therefore, it will span from July 2022 to November 2022.

4) Phase 4: Quality Assurance

- Objective: Assure the program complies with making the user a profit, that the program executes the trade securely, and that a user will not have to be an expert to use the program.
- Approach: Use a paper trail to confirm the user makes a profit and test it within the team or have the mentor comment on the ease of using the product.
- Result: A program and algorithm that complies with all objectives outlined in the report.

The Quality Assurance phase is the last check done on the product; it will ensure the product meets all objectives; if not, the team will resolve the issues until it does. It will span from September 2022 to December 2022.

These phases were created based on the work breakdown structure and outline the objectives, approaches, and results that each phase requires. The Phases section also documents the deadlines for each phase that will result in a fully working product by the end of the project's deadline.

D. Milestones

Setting milestones for the project can be rewarding for members as they are completed. It also ensures that the project is on schedule to avoid missing the deadline for the project. Our project has five milestones:

1. Research Phase - July 20, 2022
2. Library Developed – July 29, 2022
3. Algorithm Developed - October 20, 2022
4. Program Developed – November 21, 2022
5. Product Usable December 5, 2022

The Research Phase is a vital part of the project, as it sets the course for the rest of the product development. We must take great care to research how we can implement the Stock Market Trader Bot and complete it by July 20, 2022. Having the library developed by July 29, 2022, will allow us to show what the algorithm will do to analyze stocks for the final demonstration. Developing the algorithm will take considerable time, but having it done by October 20th will allow us to have time to fine-tune the algorithm as well as implement it into the program. The Program being developed by November 21st, 2022, will allow us to have a prototype and enough time to fix other issues with the product. The team will aim to finish the product before finals week; this will allow the undergraduate team members to have time to focus on other work or to keep fine-tuning the product after finals.

E. Gantt Chart

The Gantt Chart is a critical aspect of planning for a project; it will outline each task, distribute them among each member, and set a deadline for each task. By doing this, we can visually see how the distribution of tasks and ensure each member does an equal amount of work. In addition, the Gantt Chart, shown in Fig. 34 and Fig. 35, shows how the tasks will flow during the rest of the project's timeline.

		Name	Duration	Start	Finish	Predecessors
1		**Stock Market Trader Bot**	**138 days?**	**5/26/22 8:00 AM**	**12/5/22 5:00 PM**	
2		**Research**	**40 days?**	**5/26/22 8:00 AM**	**7/20/22 5:00 PM**	
3	✔	Machine Learning and Finance	18 days	5/26/22 8:00 AM	6/20/22 5:00 PM	
4	✔	Legalities	18 days	5/26/22 8:00 AM	6/20/22 5:00 PM	
5		Design Selection	15.5 days?	5/26/22 8:00 AM	7/20/22 5:00 PM	
6		Project Report	16.583 days?	5/26/22 8:00 AM	7/19/22 5:00 PM	
7		**Algorithm Development**	**66 days?**	**7/21/22 8:00 AM**	**10/20/22 5:00 PM**	**2**
8		Library	7 days?	7/21/22 8:00 AM	7/29/22 5:00 PM	
9		Machine Learning Model Development	66 days?	7/21/22 8:00 AM	10/20/22 5:00 PM	
10		IB Input	66 days?	7/21/22 8:00 AM	10/20/22 5:00 PM	
11		**Program Development**	**88 days?**	**7/21/22 8:00 AM**	**11/21/22 5:00 PM**	**2**
12		User-Friendly GUI	88 days?	7/21/22 8:00 AM	11/21/22 5:00 PM	
13		Auto/Manual Mode	88 days?	7/21/22 8:00 AM	11/21/22 5:00 PM	
14		Algorithm Implementation	22 days?	10/21/22 8:00 AM	11/21/22 5:00 PM	9
15		Trade Capability	88 days?	7/21/22 8:00 AM	11/21/22 5:00 PM	
16		**Quality Assurance**	**10 days?**	**11/22/22 8:00 AM**	**12/5/22 5:00 PM**	**2;7;11**
17		User Friendly Testing	10 days?	11/22/22 8:00 AM	12/5/22 5:00 PM	
18		Profitability (Paper Trail Testing)	10 days?	11/22/22 8:00 AM	12/5/22 5:00 PM	

Fig. 34. Task View Gantt Chart

Fig. 35. Task-Time Distribution Gantt Chart

F. Pert Chart

The Pert chart will allow the team to see how the tasks lead towards the final goal and help the organization of the tasks. Furthermore, the PERT chart is utilized in making an effective outlining of the aimed trajectory of the end product. This trajectory is critical because it is an important factor in organizing the scheduling of the implementation phase of the Stock Market Trader Bot. The following figure, Fig. 36, shows the Pert chart for our project that effectively demonstrates the designing of the planned implementation of the project.

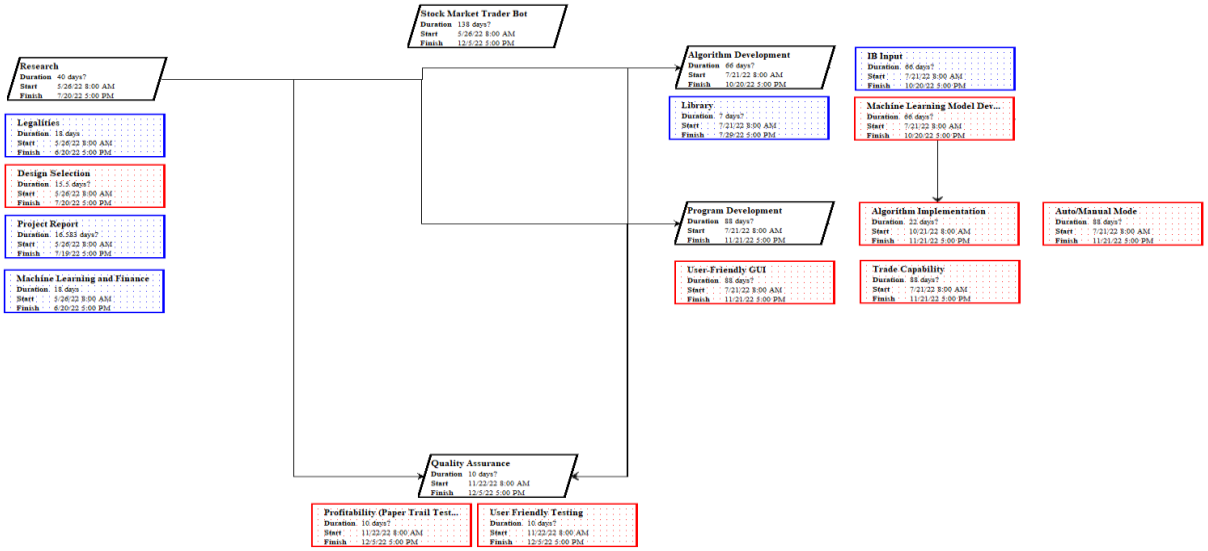


Fig. 36. PERT Chart

In conclusion, planning of action for any project is critical, as it will hold each member responsible for their tasks and keep the team on track with what needs to be done at any time. In addition, the Gantt Chart and Pert Chart allow for visualization of the flow of the project's task. Having milestones and phases for members to complete motivates the team to complete their work. Deliverable due dates must be followed strictly. However, the other due dates are not as strict but should be followed to prevent the team from falling behind.

XX. MULTIDISCIPLINARY ASPECTS

By definition, a project that is classified as multidisciplinary is one that integrates several fields of study or academic fascinations. It is important to have a team with expertise in various fields of interest to promote a well-rounded and balanced final product. More specifically, this team promotes multidisciplinary importance by forming a group of talented undergraduate engineers of both Electrical and Computer Engineering with diversified individual specialties such as artificial intelligence, embedded software, cybersecurity, machine learning, and more. The team of participating engineers of this project understands there is heightened significance behind constructing a multidisciplinary team. This section aims to effectively demonstrate how this team's multidisciplinary aspects will facilitate a swift design and implementation process for the product and how each individual member's skillset will contribute to the overall construction of a multidisciplined team and produce a successful Stock Market Trader Bot.

Each singular team member has differences in skills possessed. Furthermore, an effective team will assess said differences and coordinate how each member can contribute to the best of their abilities. After this coordination, the engineering team will have a clear picture as to how these differences in specialties and expertise can be molded to complement each other to effectively produce the aimed end product. However, before entering the subject of how each individual skillset complements one another, the proficiency and capability differences amongst members will be listed:

1) *Daniel Leal*
   - Proficient in low-level (C/C++) and high-level programming languages (Python, C#).
   - Experienced in multiple machine learning models (Support Vector Machines, K-Nearest Neighbors, Deep Learning) and datasets (Regression & Classification).
   - Various years of experience in multidisciplinary R&D environments.
   - Relevant work and numerous projects in data analysis and data science fields.
   - Skilled in varying scripting languages producing high quality user interfaces.
   - Performed valuable research for the CREPES research project at FIU's College of Engineering.
2) *Gregory Hill*
   - Over 20+ years' experience working in an engineering environment and multidisciplinary teams.
   - Relevant work experience in the fields of Sales, Satellite, Project, and Solutions Engineering.
   - Previously certified in related fields of automation and programmability through an Information Technology (IT) certification from Cisco Systems (CCNA).
   - Skilled in embedded system software and developmental boards such as Zynq and Arduino.
   - Understanding of how effective engineering projects and teams should be managed as a result of varying engineering experiences.
3) *Joseph Dabreu*
   - High quality work experience in algorithmic development pertaining to API miscommunications.
   - Proficient in performing complex LINQ queries on backend servers using C#.
   - Hands-on volunteer work as an Information Technology (IT) agent.

- Deep understanding of a heavily used hardware programming language (VHDL).
- Heightened comprehension on how to perform tests on JSON object responses from REST APIs.
- Vast apprehension of multiple IDE's and technologies such as Swagger UI, Code Blocks, Eclipse, and more.

4) Eddie Lezama

- Skilled object-oriented programmer with numerous sophisticated programming and power systems projects.
- Experience working with Java and version controls such as GitHub to make GUIs and a house adventure game.
- Intense understanding of troubleshooting procedure as a result of active work experience as an IT Assistant for a highly respected university.
- Heightened expertise in Electrical Engineering principles and computer hardware.
- Complex understanding of laptop and pc specifications as a result of work experience in upgrading technological parts of systems to improve performance.
- Possess varying technical skills such as MS apps, XAMN, XRY, and FTK Imaging.

Having teammates with many different technical backgrounds can cause for conflicting ideals. However, this team has proven to do the contrary in this engineering project. This team has combined their skills to work effectively together and complement one another in the development of the Stock Market Trader Bot. Daniel's experience in research and development, machine learning models, seven plus programming languages, and data science can be easily applied to the historical data analysis code and algorithms to be used in the project. Gregory's automation and programmability skills along with experience in an engineering atmosphere can ensure that the structure of our team is best fit for a successful end product. Joseph's experience working with algorithmic development and IT services is sure to be an extremely useful resource when it comes to creating a profitable trading algorithm for the Bot and troubleshooting any issues presented. Eddie's heightened interests and experiences with object-oriented programming and graphical user interfaces (GUIs) through Java will promote an easy integration of a friendly user environment. It is undeniable that all these different skillsets complement each other with respect to the implementation and design of the Stock Market Trader Bot and will be put to efficient use.

To minimize delays and promote leadership managerial accountability, the team will host a weekly 15-minute to 45-minute-long meeting amongst team members to update one another on the progress of individual prospective assignments. This will consistently be enforced through the entirety of the design and implementation of the Bot, where the break between semesters will be no exception. Furthermore, to ensure appropriate planning of leadership management, a formal document will list all expected individual contributions for each team assignment until the completion of the project.

In conclusion, the multidisciplinary team in charge of this project understand the importance of utilizing every participating Electrical/Computer Engineer's mastery to further extend the encouragement of a well-rounded, multidisciplined final product. Although clear differences in technological expertise is evident, these multidisciplinary aspects are seen as highly beneficial where coherency will be promoted to ensure these differencing skillsets and abilities will complement one another and allow for the advancement of a successful and profitable end product.

XXI. PERSONNEL

Daniel Leal

(786) 461-3308 | dleal016@fiu.edu | Miami, FL | lealdaniel.net | LinkedIn

Experience

Beckman Coulter Diagnostics
Software Engineering Intern, R&D June 2022 - Present
▪ Created data analysis scripts with intricate UIs through high-level languages (Python, C#, Perl) that promoted less manual labor and increased productivity of data extractions performed by development scientist by over a tenfold (1000%).

Florida International University College of Engineering
Machine Learning Researcher Jan 2022 – May 2022
▪ Perform valuable research on classification and regression datasets for the CREPES project as an undergrad researcher to perform data analytics to produce over 96% accuracy using Python and a variety of machine learning models such as SVM, Random Forest, K-NN, Deep Learning, etc.

Miami-Dade County Public Schools
Liaison Specialist Mar 2021 – May 2022
▪ Manage large data analytics / database of the entire center, automated emailing system, and promoted to full-time to manage district-wide website using WMS in efforts to maintain a friendly graphical user interface for online visitors

Education

Florida International University – Expected Graduation: Dec 2022
Bachelor of Science in Computer Engineering
▪ Concentrations: Artificial Intelligence and Embedded Software
▪ GPA: 3.75 out of 4.00

Miami Dade College – May 2020
Associate Degree in Mathematics/Computer Science
▪ GPA: 3.85 out of 4.00

Skills

Programming Languages: Python, C, C++, C#, Java, JavaScript, Perl, VHDL, CSS, HTML, Ruby, MATLAB
Tools: Computer Vision, Scikit-learn, Git, GitHub, Bootstrap, Django, Selenium, Pandas, Hive, Anaconda, Azure DevOps
Software/Hardware: MS apps, Wolfram Mathematica, Xilinx Vivado, Multisim, Arduino, Raspberry PI, Zynq ARM Controllers, FPGA, NIOS II Controllers
Interests: Artificial Intelligence, Data Science, Embedded Systems, Software Engineering
Language Fluency: Spanish and English

Gregory Hill

(954) 907-7389 | ghill001@fiu.edu | Miami, FL

Experience

Private Global Telecommunications Provider
Senior Solutions Engineer Nov 2019 - Present
▪ Offers bespoke technical solutions and in-depth expertise during customer meetings and contract negotiations and also represents the company during regional trade conferences and exhibitions.

Public Cruise Line
Satellite Engineer July 2019 – Nov 2019
▪ Serves as a liaison between Cruise company and VSAT provider, ensuring proper design, implementation, and operations of satellite communication systems.

Private Satellite Communications Provider
Director Oct 2017 – Feb 2019
▪ Lead a group of talented sales and project engineers to produce success in all aspects of the Sales and Project Engineering de.

Solutions Architect Sept 2015 – Oct 2017
▪ Developed overall technical plan and vision for satellite communication problems by performing design, debug, and performance analysis on solutions.

Education

Florida International University – Expected Graduation: Dec 2022
Bachelor of Science in Computer Engineering
▪ Concentrations: Embedded Systems and Cybersecurity
▪ GPA: 2.95 out of 4.00

Certifications

Cisco Certified Network Associate (CCNA)
Six Sigma Yellow Belt (ICYB)
Sea Tel Installation and Maintenance (STIM)
Indirect Institute for Organization Management (IOM)
Global Very-Small-Aperture Terminal Forum (GVF)

Skills

Programming Languages: Java, VHDL, C
Software/Hardware: MS apps, Xilinx Vivado, Multisim, Arduino, Zynq ARM Controllers, FPGA,
Interests: Automation, Programmability, Embedded Systems, Cybersecurity
Others: Management, Sales, Communication, Leadership

Joseph Dabreu

(305) 323-5981 | jdabr003@fiu.edu | Miami Beach, FL

Experience

ION 247 Managed IT Services
Junior Software Developer Intern June 2021 – Mar 2022
▪ Performed complex C# LINQ queries on company's backend server to obtain relational data and develop an algorithm using the relational data to prevent revenue loss due to API miscommunications.
▪ Utilized Postman and Swagger UI to perform GET requests on Datto API endpoints to develop unit tests while also performing unit tests for JSON object responses from a REST API, utilizing xUnit.net for testing.

Mount Sinai Medical Center
IT Service Agent Volunteer Jan 2022 – May 2022
▪ Reported directly to Technical Service Manager for special projects and aided with hardware troubleshooting, installation, configuration, repair, and upgrade of PC workstations, and laptops while implementing preventative maintenance rounding and inventory of hospital workstation.
▪ Implemented preventative maintenance rounding of hospital conference rooms and performed PC and laptop Reimaging for Windows 7 & 10.

Education

Florida International University – Expected Graduation: Dec 2022
Bachelor of Arts in Computer Engineering
▪ Concentrations: Networking and Cybersecurity
▪ Member of Upsilon Pi Epsilon
▪ Hackathon Participant
▪ GPA: 3.14 out of 4.00

Miami Dade College - 2019
Associate of Arts in Computer Engineering
▪ GPA: 3.54 out of 4.00

Skills

Programming Languages: Python, C, C#, VHDL, HTML, MATLAB, Mathematica
IDEs: Code Blocks, Microsoft Visual Studio, Visual Studio Code, Eclipse, Android Studio, Xilinx ISE
Software: Swagger UI, Postman, Slack, MS Teams, MS Outlook, Autotask, Datto RMM
Interests: Artificial Intelligence, Networking, Cybersecurity
Version Control: Git, Git Bash, GitHub, GitLab

Eddie Lezama

(786) 542-7231 | eleza005@fiu.edu | Miami, FL

Experience

Florida International University College of Medicine
IT Assistant Oct 2021 – Mar 2022
▪ Assisted in setting up devices for lecture recording so the class can run smoothly while being recorded for online viewers.
▪ Edited PDFs, PowerPoints, and Word Documents to make them accessible and satisfy Ally standards
▪ Replaced and upgraded parts of laptops to improve their performance.

Education

Florida International University – Expected Graduation: Dec 2022
Bachelor of Science in Electrical Engineering
▪ Concentrations: Data System Software and Digital Forensics
▪ GPA: 3.20 out of 4.00

Projects

Power Systems
Circuit Model for Pulse Motor
▪ Utilized Multisim software to make a pulse motor design that would function appropriately.
▪ Constructed the pulse model by looking for parts that would fit and scrapping old electronics to fit criteria and subsequently made the working pulse motor.
▪ Testing the pulse motor's function by spinning the magnetic disc that worked as the motor and attaching a battery, findings concluded motor can work as a pulse motor or generator

Java
Capture the Flag
▪ Designed an algorithm and converted to Java code by using pseudocode and flowcharts.

House Adventure Game
▪ Collaborated and used proper Javadoc documentation through GitHub to make a house adventure game with all methods, classes, and parameters documented for easy understanding

Tic-Tac-Toe
▪ Programmed a GUI by implementing the Javax library which resulted in GUI popups that did I/O and functioned as graphics for Tic-tac-toe

Skills

Programming Languages: C, Java, HTML, CSS, Wolfram, VHDL
Software: Power World, NI Multisim, Xilinx Vivado, MS Word, MS Excel, MS PowerPoint, XAMN & XRY, FTK Imager
Others: Object Oriented Programming (OOP), Circuits, Power Systems, Graphical User Interfaces (GUIs)
Interests: Artificial Intelligence, Data System Software, Digital Forensics
Languages: English and Spanish

XXII. BUDGET

Maintaining a proper budget is essential to a project's success. Without a budget for a project, there will be a lack of support and the project will likely falter due to improper or insufficient budget management. “A project budget is the total projected costs needed to complete a project over a defined period of time” [39]. This is used to ensure that all phases of the project will be evaluated on cost-based needs. General factors of the project budget consist of labor cost, material cost, and operating costs but many more options can be considered as the project develops. A budget not only ensures the success of a project, but also allows stakeholders to realize how much money is required and when it is needed for implementation. The budget is also a baseline that measures project performance as collection of funds progresses throughout the project when it begins.

Creating a project budget contains many components to take into consideration. Direct and indirect costs, fixed and variable costs, labor and materials, travel, equipment and space, licenses, and anything else that may arise when the project is being built. Labor cost refers to the spending required for resources working on the project. This includes anyone currently working on the project, someone being hired to work on the project, or potential hires as well. Material costs refers to expenses for materials used during the production and manufacturing of the project and typically relates to specific tools required for the execution of the project. Software costs include all expenses needed for providing team members with necessary software licenses for project management. To meet the financial aspects of a project the budget must be thorough, informative, and not missing any aspects that are required for funding. Anticipating costs is imperative when considering a project budget as many projects fail to estimate their budget due to potential costs not being estimated. As a result, funds may have to be quickly pulled from a resource to adhere to the cost that was not predicted.

When it comes to the team’s project, the stock market trader bot, the team will deal mostly with software constraints and the labor of individual team members to take into consideration. We also must factor in how long it takes to learn how the stock market operates, how to use python, and how trading algorithms can be applied to our project to produce a desired result. Learning how the bot is to be designed and the functions it will produce consists of a Udacity course on machine learning for trading. With the knowledge gained from said course, the team will be able to produce a software bot that can predict outcomes of future stock prices by algorithmically analyzing past data. According to the ProjectLibre project design file, Gantt and PERT charts, the estimated cost of the project will be $62469.96 due to the project being software labor intensive and requiring no hardware or material cost to be produced. Through each iterative phase of the project’s process, the team will be learning and implementing code based on the Udacity course in a timely manner. Thus, there will be no outstanding cost to the project itself. Research and development of the project is another matter. Expenses to take into consideration are books the Udacity course recommends. As a result of the ProjectLibre design file, the following budget has been produced:

1	Name	Cost
2	**Stock Market Trader Bot**	$62,469.96
3	**Research**	$9,069.96
4	Machine Learning and Finance	$3,060
5	Legalities	$2,160
6	Design Selection	$1,860
7	Project Report	$1,989.96
8	**Algorithm Development**	$16,680
9	Library	$840
10	Machine Learning Model Development	$7,920
11	IB Input	$7,920
12	**Program Development**	$34,320
13	User-Friendly GUI	$10,560
14	Auto/Manual Mode	$10,560
15	Algorithm Implementation	$2,640
16	Trade Capability	$10,560
17	**Quality Assurance**	$2,400
18	User Friendly Testing	$1,200
19	Profitability	$1,200

Fig. 37. Task view Budget

1	Name	Hours/Use	Cost Per Unit	Total	Grand Total
2	Daniel Leal	1,052	$15.00	$15,780.00	$62,469.96
3	Gregory Hill	943	$15.00	$14,145	
4	Joseph D'Abreu	976	$15.00	$14,640	
5	Eddie Lezama	956	$15.00	$14,340.00	

Fig. 38. Resource Cost

XXIII. RESULTS EVALUATION

Evaluating the product results allows us to determine how well the product performs, examine the team's performance, and identify what we could improve. By assessing our results, we can also ensure that our project aligns with the project's objectives and constraints. While evaluating what objectives/constraints we have met so far, we will also list the actions we can take to remedy any violations of the objectives/constraints. In addition, this section will also examine the standards we wish to align with, along with all the team outlined steps determined as necessary to avoid infringing patents.

Currently, we do not have the final product. However, we have completed a library of the functions in python that the algorithm will use. These functions utilize the pandas library and prove useful in financial analysis. The python code in our library is over 700 lines and includes Unit Tests that will be used to evaluate the results.

*A. **Objectives***

1) Safety

We only have a library for use by the algorithm so far; therefore, as our project stands, it is still safe to use by all users. There is no way for a user, no matter their experience, to be harmed by using the library.

Remedy: Our team will ensure that the program encapsulates all data transmitted via SSL encryption and program an Emergency Stop Button that stops the product from trading

2) Progressive Training

We only have a library, not an algorithm yet. Progressive Training involves having the bot use machine learning to predict future stock performance and analyze market behavior. We do not have an algorithm that incorporates machine learning, capable of predicting future stock performance and analyzing market behavior.

Remedy: Our team will develop a machine learning algorithm capable of predicting stock performance and analyzing the market; with this, the project will meet the Progressive Training objective.

3) Marketability

Marketability means having a profitable bot. We only have a library that transforms data, manipulates data, graphs data, and selects the highest valued portfolio.

Remedy: When our team creates the machine learning algorithm, it will be capable of performing trades on the market. We will also incorporate a paper-trail system to help train the bot to ensure it meets the Marketability objective. It may take many iterations of the bot trading, but our team will ensure the bot is profitable in the end product.

4) User-Friendly

User Friendly means having a program that incorporates the algorithm that interacts with the user, allowing the user to use the bot without requiring mountains of experience. This objective requires that the bot is easily upgradeable, modular, has two modes of operation: manual and automatic, and allows for easy program navigation. Our project currently does not have a program made available to a user.

Remedy: When our team develops the program incorporating the algorithm, we will have testing and feedback available to ensure that the program meets the User-Friendly objective. In

addition, we plan to make our program work in the future, meaning we will be able to analyze new stocks coming into the market, which can accomplish the upgradeable part of this objective. Making several options available to the user is a vital part of the program we will develop.

*B. **Constraints***

1) No material cost and work in standard computers

Because our project is more software-based and does not directly include hardware aspects, our material cost is 0$. In addition, because Python can be used in most operating systems, it can be used in standard computers without much issue. However, we do not currently have a functioning product, and therefore:

2) Cross-Platform

We can ensure that the product will be cross-platform by using Python for all programming of our project. However, it may require python packages to be installed on specific platforms.

3) Possess the ability to act as a simulator in the trading market

We do not have a working algorithm or paper-trail system developed yet; however, we will meet this constraint once these are complete.

*C. **Standards***

1) 12207-2017 - ISO/IEC/IEEE International Standard

This standard requires organizing the software into a hierarchical manner where a Software System is composed of software system elements. To achieve this standard, the team will keep the software organized as stated by the standard.

2) ISO/IEC/IEEE 14764 - International Standard

To achieve this standard, the team must maintain a certain maintenance process described in the Standard Considerations section.

3) ISO/ IEC/IEEE 29119-1

To achieve this standard, the team will have a verification and validation process to maintain the software.

*D. **Avoiding Infringing Patents***

1) System and Method for a Risk Check

To avoid infringing this patent, we will avoid using a Risk Value and instead predict the stock market's future to determine what trades would be the most profitable. Our Stock Market Trader Bot will not be infringing on this patent.

2) Methods, Systems, and Computer Program Products for Trading Financial Instruments on an Exchange

3) System and Method for Order Placement in an Electronic Trading Environment

These patents concern the handling of trade transactions, not predicting the stock market or executing trades, so our Stock Market Trader Bot will not be infringing on these patents.

*E. **Specifications***

1) Secure transactions

Transmitted data is encapsulated via SSL encryption

2) Safe Trading

The product includes an Emergency Stop button to stop all trading

3) Operating Systems

Product compatible with Microsoft (Windows 10 or newer), macOS, and Linux

4) Memory Requirements

The product requires at least 512 KB to install, 128 KB RAM

5) Accuracy

The program has more than 60% accuracy in predicting the stock market

6) Ease of use

Less than 2 hours of learning is required for new users and novice traders

7) Profitable Trading

$\alpha > 1$, the algorithm outperforms S&P 500

F. ***Conclusion***

The evaluation method will accurately and honestly assess the project's results. It will be based on the objectives, constraints, standards, and specifications listed that the project meets and if the project avoids infringing on said patents. The team will ensure that the project's evaluation will be satisfactory.

XXIV. LIFE-LONG LEARNING

The project's primary goal is to create an accurate Stock Market Trading Bot. This endeavor may create multiple future opportunities and impact our lives. This project will take hours upon hours of programming work to complete correctly. In addition, each member needs to put the most effort into bringing the product to life. While we do so, it is also essential for us to learn as much as we can from each other and other sources such as IEEE.

Developing any machine learning algorithm in addition to a program that can trade on the stock market is no easy task. This is why it requires much programming labor to complete the final product while accomplishing the objectives. In addition, we must implement a machine learning model that will result in profitability for our users. The project also requires that there are only a limited number of users. This is because the product will be ineffective if too many users are profiting using the same portfolios, resulting in the product being not profitable. However, this project is also beneficial for the team as it provides experience in financial analysis and may help us find jobs in hedge funds to make trading bots for them. The project also opens up the machine learning field for most members and provides a machine learning experience for all members.

The team also holds the option to keep developing the product in the future. By doing so, we can keep learning about artificial intelligence and the stock market far into the future. In addition, we can also expand the trading capacity to include volatile financial instruments such as cryptocurrency.

There are many materials and organizations to help us with this project. Organizations such as IEEE, AAAI, ACET, and hedge funds may provide materials to help us better understand or collaborate with others who have relevant experience developing a Stock Market Trader Bot. I am subscribed to IEEE's magazine and am a member of IEEE. The organization has been beneficial in exploring not only machine learning but other topics as well. They also allow me to contact other engineers within the society that can help expand my network.

In summary, the continuation of the project will keep the team engaged during the learning process. The team is committed to having each member explore and help as much as possible on the project. Once the project is complete, we will have a machine learning experience we can add to our resumes, in addition to helping us to obtain careers within hedge funds. We can also expand the project to include more financial instruments than stocks. Relying on organizations such as IEEE can help us learn or overcome challenges that may arise from developing our product.

XXV. CONCLUSION

Prior to the opportunity of a team-oriented project implementation, the participating engineers of this team have always possessed heightened interest in composing a product that involved the opportunity for financial profitability whilst implementing principles of computer and electrical engineering undergraduate studies. It is purely because of this common motivation and goal; the Stock Market Trader Bot design and implementation phases have begun. In the design phase of the Bot, the main objectives include safety, progressive training, marketability, and user friendliness. More specifically, these main objectives specifically quantify that the end product aims to ensure the product will have a success measurement high enough for the product to be generalized as profitable, it will have the ability to opt-in and provide two functional options for trading (Automatic Transaction Handling or Client/User Intervention), and it will be easy to utilize and maneuver by the user with minimal prior trading and stock exchange background. The idea of a high profit measurement to be instilled as well as confirming the Bot has a friendly user interface making it easy to maneuver with minimal trading experience resulted from the client interview performed. The last of the mentioned objectives, having two options for functional trading (Automatic or Manual), was composed as a direct result of team member brainstorming as an envision of a widespread, easily usable final product. Furthermore, the composition of this design phase and proposal with respect to the Stock Market Trader Bot involved a variety of activities to promote a swift and organized completion. More specifically, the activities involved in completing this proposal includes team member brainstorming of potential projects, consulting with the team mentor regarding the best overall project, the steps in order to understanding all aspects of the project, background and in-depth research of the aimed end product, client interview for important objectives of the final product, feasibility analysis, and end-product deliverables. As described in the prior Results Evaluation section, objectives, constraints, standards, and avoiding patent infringement is essential to the compilation of the Bot, Overall, the main goal for the product is to provide a high chance of gaining pecuniary profit with minimal prior experiencing by the end of the implementation phase (Senior Design II) of the Stock Trader Bot. To do so, along with the main objectives, constraints must be met such as possessing the capability to operate as a simulator for potential transactions, comparability with standard computers, and ensuring cross-platform abilities. Lastly, as outlined in the results evaluation and deliverables section, the Stock Trader Bot will be designed and implemented in a manner that complies with all AI and IEEE standards for proper publication. By following through with all claims and goals outlined in the prior sections, the project is sure to positively contribute to society in a global manner by providing its users monetary earnings with little to no applied effort. In conclusion, the entirety of the process pertaining to the designing of the Stock Market Trader Bot has contributed to significant lifelong learnings for all team members in fields of high demand such as machine learning, artificial intelligence, software development in a team environment, economics and finance, and data analytics through statistical and programming applications.

XXVI. REFERENCES

[1] B.M. Barber, T. Odean, *Do Day Traders Rationally Learn About Their Ability*, University of California, Berkeley, October 2017, pp. 3, from https://faculty.haas.berkeley.edu/odean/papers/Day Traders/Day Trading and Learning 110217.pdf.

[2] M. Lyck, *Why 80% of Day Traders Lose Money*, Stock Platform Etoro, June 2020, from https://marklyck.medium.com/why-80-of-day-traders-lose-money-78d51b10fe25.

[3] A. Hayes, M.J. Boyle, R. Eichler, *What is a Mutual Fund?*, Investopedia, March 2022, from https://www.investopedia.com/terms/m/mutualfund.asp.

[4] Finra, *Exchange-Traded Funds*, Finra.org, June 2022, from https://www.finra.org/investors/learn-to-invest/types-investments/investment-funds/exchange-traded-fund.

[5] G.T. Lins, T.P. Lemke, K.L. Hoenig, *Hedge Funds and Other Private Funds: Regulation and Compliance*, P.S.Rube, September 2014.

[6] B. Beers, C. Potters, K. Munichiello, *Where Can I find Historical Stock/Index Quotes?*, Investopedia, June 2021, from https://www.investopedia.com/ask/answers/find-historical-stock-index-quotes/.

[7] T. Balch, A. Chakraborty, *Machine Learning for Trading*, Udacity & Georgia Tech, June 2022, from https://www.udacity.com/cthese/machine-learning-for-trading--ud501.

[8] B.O. Montreal, *How does breaking news affect markets?*, Bank of Montreal, May 2020, from https://www.bmo.com/main/personal/investments/learning-centre/how-breaking-news-affects-markets/#:~:text=How%20does%20news%20impact%20stock,stock%20prices%20will%20be%20affected.

[9] Fidelity, *Bollinger Bands*, Fidelity, June 2022, from https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/bollinger-bands#:~:text=Bollinger%20bands%20help%20determine%20whether,signals%20given%20with%20other%20indicators.

[10] R.O. Michaud, D.N. Esch, *The Fundamental Law of Active Management*, New Frontier Advisors, LLC, June 2017, pp. 2, from https://newfrontieradvisors.com/media/1376/fundamental-law-june-2017.pdf

[11] W. Kenton, J. Mansa, S. Kvilhaug, *Capital Asset Pricing Model (CAPM)*, Investopedia, January 2022, from https://www.investopedia.com/terms/c/capm.asp#:~:text=The%20Capital%20Asset%20Pricing%20Model%20(CAPM)%20describes%20the%20relationship%20between,assets%20and%20cost%20of%20capital

[12] Robeco, *Capital Asset Pricing Method (CAPM)*, Robeco The Investment Engineers, June 2022, from https://www.robeco.com/en/key-strengths/quant-investing/glossary/capital-asset-pricing-model.html.

[13] K. Wharton. *How the Father of Arbitrage Pricing Theory Influenced Wall Street*, April 2020.

[14] Investment Management Lab, *What is the Arbitrage Pricing Theory?*, Investment Management Lab, May 2021, from https://www.youtube.com/watch?v=wZsPfHUt6xs.

[15] M. Scarpino, *Algorithmic trading with interactive brokers: Python and C++*, Fishers, IN: Quiller Technologies LLC, 2019, pp. 8.

[16] Wikipedia, *2021 Form 10-K*, Interactive Broker, February 2022

[17] F. (n.d.). *Finnhub - Free real-time APIs for stock, forex, and cryptocurrency*, Retrieved June 14, 2022, from https://finnhub.io/.

[18] M. Carr, G. Scott, *Turtle Trading: A Market Legend*, Investopedia, August 2021, https://www.investopedia.com/articles/trading/08/turtle-trading.asp#:~:text=Turtle%20Trading%20is%20based%20on,most%20famous%20trend%2Dfollowing%20strategies.

[19] Finnhub - Free Stock API for Investors. (2020, November 18). Kickstarter. Retrieved June 14, 2022, from https://www.kickstarter.com/projects/finnhub-stock-api/finnhub-free-stock-api-for-investors/description.

[20] M. Scarpino, *Algorithmic trading with interactive brokers: Python and C++*, Fishers, IN: Quiller Technologies LLC, 2019, pp. 604.

[21] Calderon, P. C. (2019, July 22). *Automated stock market trading software to generate wealth*. Retrieved June 13, 2022, from https://www.kickstarter.com/projects/maverickbusiness/automated-stock-market-trading-software-to-generate-wealth/description

[22] S. P. Mintz, *System and Method for a Risk Check*, United States Patent 11,216,880 Apr. 30, 2022.

[23] G. L. Gastineau, T. J. Broms, D. J. McCabe, P.E. Kuhnle, *Methods, Systems, and Computer Program Products for Trading Financial Instruments on an Exchange*, United States Patent 7,496,531 Mar. 7, 2007.

[24] M. J. Burns, E. M. Herz, S. P. Mintz, A. D. Dietz, *System and Method for Order Placement in an Electronic Trading Environment*, United States Patent 8,712,904 Feb. 7, 2013

[25] M. Scarpino, "Chapter 6 Fundamental Classes of the TWS API," in *Algorithmic trading with interactive brokers: Python and C++*, Fishers, IN: Quiller Technologies LLC, 2019.

[26] O. Hegazy, O. S. Soliman, M. A. Salam, *A Machine Learning Model for Stock Market Prediction*, *IJCST*, vol. 4, no. 12, pp. 17-23, Dec. 2013

[27] A. R. Azhikodan, A. G. K. Bhat, M. V. Jadhav, *Stock Trading Bot Using Deep Reinforcement Learning*, LNNS, vol. 32, pp. 41-49, May 2018

[28] C. Nicholson, *A Beginner's Guide to Deep Reinforcement Learning*, Retrieved Jun. 16, 2022, from https://wiki.pathmind.com/deep-reinforcement-learning.

[29] T. Balch, D. Dave, *Machine Learning for Trading*, Udacity, Retrieved Jun. 17, 2022, from https://classroom.udacity.com/ctheses/ud501.

[30] A. Watters,*General Technology Statistics*, February 3, 2022, from https://connect.comptia.org/blog/....

[31] A. Watters, *AI and Machine Learning Technology Growth Statistics*, February 3, 2022 from https://connect.comptia.org/blog/.....

[32] World Trade Organization (WTO), *Who We Are,* Retrieved July 10, 2022, from https://www.wto.org/english/thewto_e/whatis_e/who_we_are_e.htm.

[33] World Trade Organization (WTO), *Understanding the WTO,* Retrieved July 10, 2022, from https://www.wto.org/english/thewto_e/whatis_e/who_we_are_e.htm.

[34] World Trade Organization (WTO), *What is the WTO,"* Retrieved July 10, 2022, from https://www.wto.org/english/thewto_e/whatis_e/whatis_e.htm.

[35] RAS/SC – Standing Committee for Standards, *IEEE Standard for Autonomous Bots (AuR) Ontology,* May 12, 2022, from https://standards.ieee.org/ieee/1872.2/7094/.

[36] A.R. Deepthi, *KNN Visualization in just 13 lines of code*, TowardsDataScience, September 24, 2019, from https://towardsdatascience.com/knn-visualization-in-just-13-lines-of-code-32820d72c6b6.

[37] Simplilearn, *What is Q-Learning…,* February 16, 2022, from https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-q-learning.

[38] Fidelity, *Bollinger Bands Description,* Retrieved July 14, 2022, https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/bollinger-bands#:~:text=Bollinger%20Bands%20are%20envelopes%20plotted,Period%20and%20Standard%20Deviations%2C%20StdDev.

[39] Bridges, J. (2019, December 3). *7 Steps for a Successful Project Budget*. ProjectManager. Retrieved July 11, 2022, from https://www.projectmanager.com/training/create-and-manage-project-budget#:%7E:text=A%20project%20budget%20is%20the,procurement%20costs%20and%20operating%20costs

[40] 12207-2017 - ISO/IEC/IEEE International Standard - *Systems and software engineering – Software life cycle processes*, November 15, 2017, IEEE Standard | IEEE Xplore, Retrieved July 12, 2022, from https://ieeexplore.ieee.org/document/8100771.

[41] 14764-2021 - ISO/IEC/IEEE International Standard - *Software engineering - Software life cycle processes – Maintenance*, January 21, 2022, IEEE Standard | IEEE Xplore, Retrieved July 11, 2022, from https://ieeexplore.ieee.org/document/9690131.

[42] 29119-1-2021 - ISO/IEC/IEEE International Standard - *Software and systems engineering – Software testing --Part 1: General concepts*, January 27, 2021, IEEE Standard | IEEE Xplore, Retrieved July 14, 2022, from https://ieeexplore.ieee.org/document/9698145.

[43] Cleancode, *Clean Code Matters*, Retrieved July 23, 2022, from https://cs.lmu.edu/~ray/notes/cleancode/.

[44] L. Golsteiin, *Life Cycle Assessment (LCA) Explained*, Retrieved July 24, 2022, from https://pre-sustainability.com/articles/life-cycle-assessment-lca-basics/.

XXVII. APPENDICES

A. Team Contract

As a member of the Team 15, I hereby agree to the following conditions:

1. I will demonstrate great interest to participate in class, share my ideas and discuss them openly with other team members.
2. I agree to follow the rules and guidelines that have been attained and established by the team in a "majority of votes" decision.
3. I am solely responsible for any assigned material by the team. I will submit my work on time and in good shape.
4. In case of an unforeseen absence, it is my responsibility to promptly contact my team members and learn of any new material. An announced and anticipated absence is much appreciated.
5. My performance is regularly reviewed and openly shared by the team. In case of a negative performance (decided by the majority of votes) I will be issued a written warning.
6. The team holds the right to release me after the third (3rd) warning (decided by the majority of votes), I am thereof entitled to file an appeal to the class professor and request arbitration.
7. Reason(s) to issue a warning may be but are not restricted by the following reasons:
    a. Unable to submit an assignment on time.
    b. Lack of team participation.
    c. Obscene and improper conduct.
8. I am not allowed to abandon my team under any circumstances.

Team Leader Name	Signature	Date	Roles
Daniel Leal	*Daniel Leal*	6/14/2022	Facilitate communication among members
			Create programming for IB and composition of statistical algorithms
Team Member Name	Signature	Date	Roles
Joseph Dabreu	*Joseph Dabreu*	6/15/2022	Review and oversee the code
			Ensure appropriate interfacing between the IB algorithm and GUI
Gregory Hill	*Gregory Hill*	6/15/2022	Create an appropriate user interface for ease-of-use
			Ensure user interface promotes facility
Eddie Lezama	*Eddie Lezama*	6/15/2022	Create code that follows mathematical models
			Create code involving financial theories and their variables

*B. **Intellectual Property Contract***

As a member of the Team 15 team, we hereby agree to the following:

1. The team, consisting of Daniel Leal, Joseph Dabreu, Gregory Hill, and Eddie Lezama has approved this contract.
2. The designated spokesman of Team 15 is Daniel Leal.
3. In case of the invention going to the market, the profit will be shared evenly amongst all the team members of Team 15.
4. Any decision regarding the intellectual property of Team 15, will be determined by majority vote between team members and mentor. All team members need to be physically present in order to hold a voting session. If a decision cannot be made by the team, the mentor will be consulted in order to make an informed decision.

Team Leader Name	Signature	Date
Daniel Leal	*Daniel Leal*	6/14/2022
Team Member Name	Signature	Date
Joseph Dabreu	*Joseph Dabreu*	6/15/2022
Gregory Hill	*Gregory Hill*	6/15/2022
Eddie Lezama	*Eddie Lezama*	6/15/2022
(Mentor) Wilmer Arellano		

C. Provisional Patent Form

PTO/SB/16 (10-20)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no persons are required to respond to a collection of information unless it displays a valid OMB control number

PROVISIONAL APPLICATION FOR PATENT COVER SHEET – Page 1 of 2

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

Priority Mail Express® Label No. _____

INVENTOR(S)

Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
Daniel	Leal	Florida
Joseph	Dabreu	Florida
Eddie	Lezama	Florida
Gregory	Hill	Florida

Additional inventors are being named on the _____ separately numbered sheets attached hereto.

TITLE OF THE INVENTION (500 characters max):

Stock Market Trader Bot

Direct all correspondence to: **CORRESPONDENCE ADDRESS**

☐ The address corresponding to Customer Number: _____

OR

☑ Firm or Individual Name: Florida International University - College of Engineering and Computing

Address: 10555 W Flagler St

City	State	Zip
Miami	Florida	33174
Country: United States of America	Telephone: (305) 348 - 2522	Email: dleal016@fiu.edu

ENCLOSED APPLICATION PARTS (*check all that apply*)

☐ Application Data Sheet. See 37 CFR 1.76.
☐ CD(s), Number of CDs _____
☑ Drawing(s) *Number of Sheets* 30
☑ Other (specify) Other Considerations
☑ Specification (e.g., description of the invention) *Number of Pages* 115

Fees Due: Filing Fee of $300 ($150 for small entity) ($75 for micro entity). If the specification and drawings exceed 100 sheets of paper, an application size fee is also due, which is $420 ($210 for small entity) ($105 for micro entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

METHOD OF PAYMENT OF THE FILING FEE AND APPLICATION SIZE FEE FOR THIS PROVISIONAL APPLICATION FOR PATENT

☐ Applicant asserts small entity status. See 37 CFR 1.27.
☐ Applicant certifies micro entity status. See 37 CFR 1.29.
Applicant must attach form PTO/SB/15A or B or equivalent.
☐ A check or money order made payable to the *Director of the United States Patent and Trademark Office* is enclosed to cover the filing fee and application size fee (if applicable).
☐ Payment by credit card. Form PTO-2038 is attached.
☐ The Director is hereby authorized to charge the filing fee and application size fee (if applicable) or credit any overpayment to Deposit Account Number: _____.

$770
TOTAL FEE AMOUNT ($)

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

This collection of information is required by 37 CFR 1.51. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 10 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/16 (10-20)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no persons are required to respond to a collection of information unless it displays a valid OMB control number

PROVISIONAL APPLICATION FOR PATENT COVER SHEET – Page 2 of 2

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government. (NOTE: Providing this information on a provisional cover sheet, such as this Provisional Application for Patent Cover Sheet (Form PTO/SB/16), does not satisfy the requirement of 35 U.S.C. 202(c)(6), which requires that the *specification* contain a statement specifying that the invention was made with Government support and that the Government has certain rights in the invention.)

☑ No.

☐ Yes, the invention was made by an agency of the U.S. Government. The U.S. Government agency name is:

_____

☐ Yes, the invention was made under a contract with an agency of the U.S. Government.

The contract number is: _____

The U.S. Government agency name is: _____

_____

In accordance with 35 U.S.C. 202(c)(6) and 37 CFR 401.14(f)(4), the specifications of any United States patent applications and any patent issuing thereon covering the invention, including the enclosed provisional application, must state the following:

"This invention was made with government support under [IDENTIFY THE CONTRACT] awarded by [IDENTIFY THE FEDERAL AGENCY]. The government has certain rights in the invention."

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

SIGNATURE Daniel Leal DATE 07/27/2021

TYPED OR PRINTED NAME Daniel Leal REGISTRATION NO. _____ (*if appropriate*)

TELEPHONE 786-461-3308 DOCKET NUMBER _____

XXVIII. SIGNATURES PAGE

Course Number: EEL 4920 Semester: Summer Year: 2022

Mentor Name: Dr. Wilmer Arellano

Senior I Instructor's Name: Dr. Wilmer Arellano

Name	PID	E-mail Address	Phone Number
Daniel Leal	6242461	Dleal016@fiu.edu	(786) 461-3308
Joseph Dabreu	5488690	Jdabr003@fiu.edu	(305) 323-5981
Gregory Hill	1155823	Ghill001@fiu.edu	(954) 907-7389
Eddie Lezama	6060915	Eleza005@fiu.edu	(786) 542-7231

	PRINT	SIGNATURE	DATE
Team Leader	Daniel Leal	*Daniel Leal*	7/25/2022
Team Member	Joseph Dabreu	*Joseph Dabreu*	7/25/2022
Team Member	Gregory Hill	*Gregory Hill*	7/25/2022
Team Member	Eddie Lezama	*Eddie Lezama*	7/25/2022
Mentor	Dr. Wilmer Arrelano		